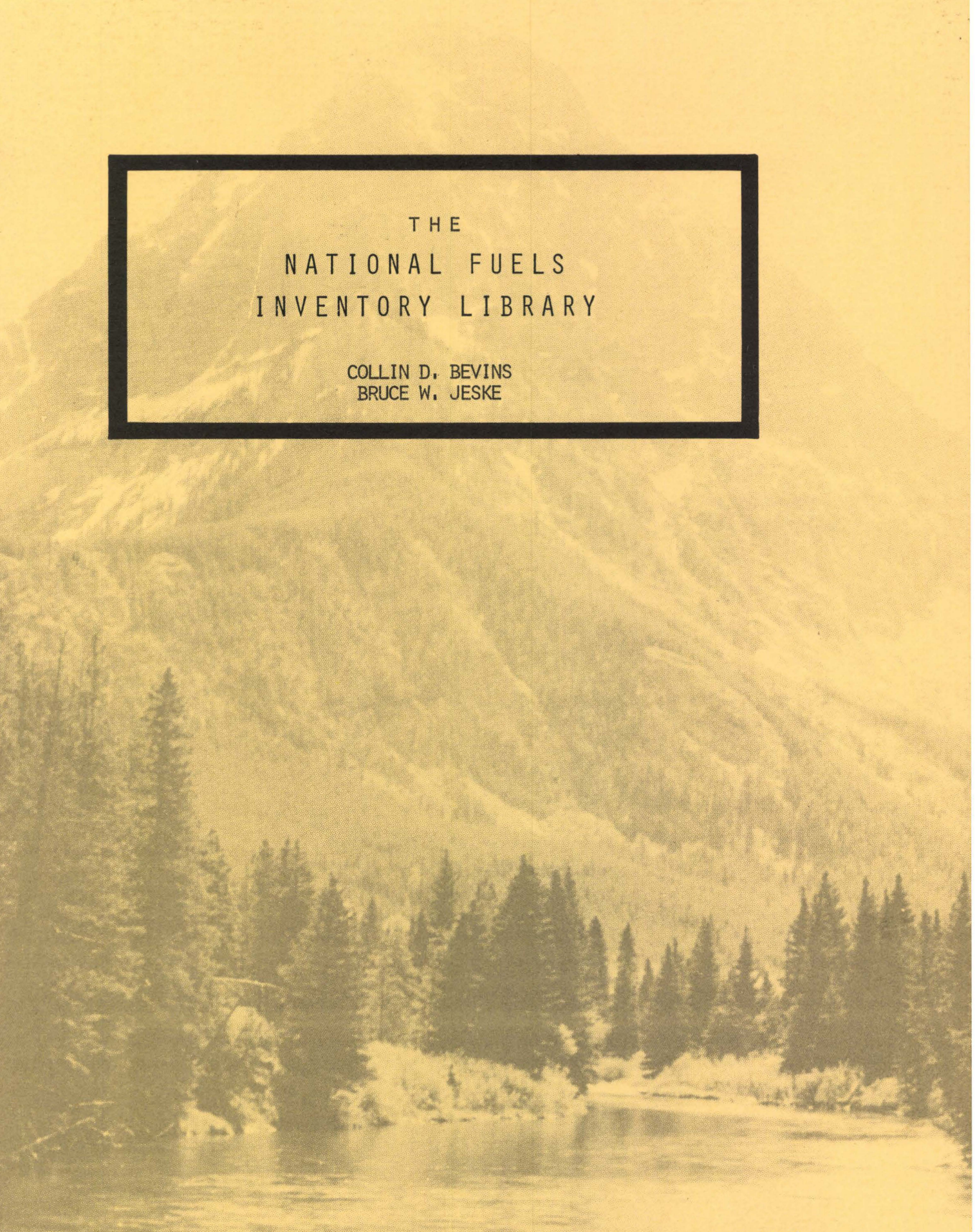
NATIONAL FUELS INVENTORY LIBRARY

SYSTEMS FOR ENVIRONMENTAL MANAGEMENT
P.O. BOX 3776
MISSOULA, MONTANA 59806

NATIONAL FUELS INVENTORY & APPRAISAL PROJECT
ROCKY MOUNTAIN FOREST & RANGE EXPERIMENT STATION
FORT COLLINS, COLORADO 80521

THE NATIONAL FUELS INVENTORY LIBRARY

COLLIN D. BEVINS
BRUCE W. JESKE



SYSTEMS FOR ENVIRONMENTAL MANAGEMENT

P. O. BOX 3776
MISSOULA, MONTANA 59806
PHONE: (406) 549-7478

LIBRARY
MAY 7 1980
ROCKY MOUNTAIN STATION

N NATIONAL

F FUELS

I INVENTORY

L LIBRARY

SYSTEMS FOR ENVIRONMENTAL MANAGEMENT
P.O. BOX 3776
MISSOULA, MONTANA 59806

NATIONAL FUELS INVENTORY & APPRAISAL PROJECT
ROCKY MOUNTAIN FOREST & RANGE EXPERIMENT STATION
U.S.D.A. FOREST SERVICE
FORT COLLINS, COLORADO 80521

0.1. PREFACE

The NATIONAL FUELS INVENTORY LIBRARY was designed and created under Cooperative Aid Agreement 16-842-CA. Cooperators on the project were:

National Fuels Inventory & Appraisal Project
USDA Forest Service, Rocky Mountain Forest & Range Experiment Station
240 West Prospect Street
Fort Collins, Colorado 80521

Stanley Hirsch, Project Leader
Peter Roussopoulos, Team Leader

and

Systems for Environmental Management (SEM)
PO Box 3776
Missoula, Montana 59806

Collin D. Bevins, Principal Investigator
Bruce W. Jeske, Research Forester

The National Fuels Inventory & Appraisal Project provided financial support and technical direction for the NFIL. Systems for Environmental Management, a Montana nonprofit research corporation operating within the public domain, designed and built the NFIL, employed and supervised personnel working on the project, and provided administrative services in support of the project.

The NFIL was designed and built by Collin D. Bevins, SEM Research Supervisor. The initial Fuels Inventory Database was built by Bruce W. Jeske, SEM Research Forester.

0.2. N F I L UPDATE PACKAGES

UPDATE NUMBER Ø:Original DATE 01 OCTOBER 1979

SECTION	DESTROY FORMER PAGES NUMBERED	FILE NEW PAGES NUMBERED

0.3. N F I L PAGE STATUS SUMMARY

UPDATE ISSUE : ORIGINAL

The following table lists the status of each page in this document and indicates the update package (if applicable).

SECTION	PAGE NUMBER	PAGE STATUS	UPDATE LEVEL
Cover	0.0.	Original	Ø
Preface	0.1.	Original	Ø
Update Packages	0.2.	Original	Ø
Contents	0.4.	Original	Ø
List of Figures	0.9.	Original	Ø
Section 1	1- 1 thru 1-17	Original	Ø
Section 2	2- 1 thru 2-40	Original	Ø
Section 3	3- 1 thru 3-136	Original	Ø
Appendix A	A- 1 thru A- 6	Original	Ø
Appendix B		Incomplete	
Appendix C		Incomplete	
Appendix D		Incomplete	
Appendix E		Incomplete	
Appendix F	F.1.	Original	Ø
Appendix G	G.1.	Original	Ø

0.4. TABLE OF CONTENTS

0.1. PREFACE 0- 1

0.2. NFIL UPDATE PACKAGES 0- 2

0.3. NFIL PAGE STATUS SUMMARY 0- 3

0.4. TABLE OF CONTENTS 0- 4

0.5. LIST OF FIGURES 0- 9

1. INTRODUCTION

1.1. PURPOSE AND STRUCTURE 1- 1

1.2. INTRODUCTION TO THE FUELS INVENTORY DATABASE (FID) 1- 4

1.2.1. Inventory Data File 1- 5

1.2.2. Stand Documentation File 1- 7

1.2.3. Variable Index File 1- 8

1.3. INTRODUCTION TO THE MANAGEMENT SOFTWARE PACKAGE (MSP) 1- 9

1.3.1. PROGRAM INDEX 1-10

1.3.2. PROGRAM DATA 1-11

1.3.3. PROGRAM COUNT 1-13

1.3.4. PROGRAM LIST 1-14

1.3.5. PROGRAM REVIEW 1-15

1.3.6. PROGRAM SEARCH 1-16

2. FUELS INVENTORY DATABASE (FID)

2.1. INVENTORY DATA FILE 2- 1

2.1.1. Purpose 2- 1

2.1.2. Structure 2- 1

2.1.3. Maintenance and Use 2- 6

2.1.4. Inventory Data File Sample Listing 2- 8

2.2. STAND DOCUMENTATION FILE 2-11

2.2.1. Purpose 2-11

2.2.2. Structure 2-11

2.2.3. Maintenance and Use 2-17

2.2.4. Stand Documentation File Sample Listing 2-19

2.3. VARIABLE INDEX FILE 2-23

2.3.1. Purpose 2-23

2.3.2. Structure 2-23

2.3.3. Maintenance and Use 2-27

2.3.4. Variable Index File Sample Listing 2-29

2.4. PERIPHERAL DATA FILE 2-32

2.4.1. Purpose 2-32

2.4.2. Structure 2-33

2.4.3. Maintenance and Use 2-38

2.4.4. Peripheral Data File Sample Listing 2-39

3. MANAGEMENT SOFTWARE PACKAGE

3.1. PROGRAM INDEX 3- 1

3.1.1. Purpose 3- 1

3.1.2. Program Input 3- 2

3.1.3. Program Output 3-10

3.1.4. Program Limitations 3-15

3.1.5. Program Error Messages 3-17

3.1.6. Program Structure and Logic 3-20

3.1.7. Logical Device Assignment 3-28

3.2. PROGRAM DATA 3-29

3.2.1. Purpose 3-29

3.2.2. Program Input 3-30

3.2.3. Program Output 3-36

3.2.4. Program Limitations 3-40

3.2.5. Program Error Messages 3-42

3.2.6. Program Structure and Logic 3-44

3.2.7. Logical Device Assignment 3-52

3.3. PROGRAM COUNT 3-54

3.3.1. Purpose 3-54

3.3.2. Program Input 3-55

3.3.3. Program Output 3-56

3.3.4. Program Limitations 3-57

3.3.5. Program Error Messages 3-59

3.3.6. Program Structure and Logic 3-60

3.3.7. Logical Device Assignment 3-62

3.4. PROGRAM LIST 3--63

3.4.1. Purpose 3--63

3.4.2. Program Input 3--64

3.4.3. Program Output 3- 68

3.4.4. Program Limitations 3- 78

3.4.5. Program Error Messages 3- 79

3.4.6. Program Structure and Logic 3- 80

3.4.7. Logical Device Assignment 3- 86

3.5. PROGRAM REVIEW 3- 87

3.5.1. Purpose 3- 87

3.5.2. Program Input 3- 88

3.5.3. Program Output 3- 90

3.5.4. Program Limitations 3- 94

3.5.5. Program Error Messages 3- 95

3.5.6. Program Structure and Logic 3- 96

3.5.7. Logical Device Assignment 3-105

3.6. PROGRAM SEARCH 3-107

3.6.1. Purpose 3-107

3.6.2. Program Input 3-109

3.6.3. Program Output 3-125

3.6.4. Program Limitations 3-132

3.6.5. Program Error Messages 3-134

3.6.6. Program Structure and Logic 3-135

3.6.7. Logical Device Assignment 3-136

A. APPENDIX A : NFIL STATUS SUMMARY A- 1

A.1. OCTOBER 1979

B. APPENDIX B : PROGRAM ACCESS & CONTROL STREAMS B- 1

C. APPENDIX C : A COMPREHENSIVE EXAMPLE OF MSP PROGRAM USE C- 1

D. APPENDIX D : MSP PROGRAM COSTS D- 1

E. APPENDIX E : RECOMMENDED VARIABLE INDEX FILE CODES E- 1

F. APPENDIX F : MSP PROGRAM LISTINGS F- 1

F.1. PROGRAM INDEX

F.2. PROGRAM DATA

F.3. PROGRAM COUNT

F.4. PROGRAM LIST

F.5. PROGRAM REVIEW

F.6. PROGRAM SEARCH

G. APPENDIX G : FID FILE LISTINGS G- 1

G.1. INVENTORY DATA FILE

G.2. VARIABLE INDEX FILE

G.3. STAND DOCUMENTATION FILE

0.5. LIST OF FIGURES

1.1. FUNCTIONAL RELATIONSHIPS BETWEEN THE FUELS INVENTORY DATABASE FILES AND MANAGEMENT SOFTWARE PACKAGE PROGRAMS 1- 1

2.1. INVENTORY DATA FILE ACCESS 2- 7

2.2. INVENTORY DATA FILE SAMPLE LISTING 2- 9

2.3. INVENTORY DATA FILE SAMPLE LISTING PRODUCED BY PROGRAM LIST 'DATA' COMMAND 2- 10

2.4. STAND DOCUMENTATION FILE ACCESS 2- 18

2.5. STAND DOCUMENTATION FILE SAMPLE LISTING 2- 20

2.6. STAND DOCUMENTATION FILE SAMPLE LISTING PRODUCED BY PROGRAM LIST 'STANDS' COMMAND 2- 21

2.7. STAND DOCUMENTATION FILE SAMPLE LISTING PRODUCED BY PROGRAM LIST 'STANDS AND PLOTS' COMMAND 2- 22

2.8. VARIABLE INDEX FILE ACCESS 2- 28

2.9. VARIABLE INDEX FILE SAMPLE LISTING 2- 30

2.10. VARIABLE INDEX FILE SAMPLE LISTING PRODUCED BY PROGRAM LIST 'INDEX' COMMAND 2- 31

2.11. PERIPHERAL DATA FILE SAMPLE LISTING 2- 40

3.1. PROGRAM INDEX 'CREATE' MODE SAMPLE INPUT STREAM 3- 4

3.2. PROGRAM INDEX 'ADD' MODE SAMPLE INPUT STREAM. 3- 9

3.3. PROGRAM INDEX SAMPLE TITLE PAGE OUTPUT 3- 11

3.4. PROGRAM INDEX 'CREATE' MODE SAMPLE OUTPUT 3- 12

3.5. PROGRAM INDEX SAMPLE CLOSING PAGE OUTPUT 3- 13

3.6. PROGRAM INDEX 'ADD' MODE SAMPLE OUTPUT 3- 14

3.7. PROGRAM INDEX ERROR MESSAGES 3- 18

3.8. PROGRAM INDEX SAMPLE CLOSING PAGE OUTPUT FOR INPUT STREAM WITH SYNTAX ERRORS 3- 19

3.9. PROGRAM DATA 'CREATE' MODE SAMPLE INPUT STREAM 3- 32

3.10. PROGRAM DATA SAMPLE TITLE PAGE OUTPUT 3- 37

3.11. PROGRAM DATA 'CREATE' MODE SAMPLE OUTPUT 3- 38

3.12. PROGRAM DATA SAMPLE CLOSING PAGE OUTPUT 3- 39

3.13. PROGRAM DATA ERROR MESSAGES 3- 43

3.14. PROGRAM LIST SAMPLE INPUT STREAM 3- 65

3.15. PROGRAM LIST SAMPLE TITLE PAGE OUTPUT 3- 69

3.16. PROGRAM LIST 'INDEX' COMMAND SAMPLE OUTPUT 3- 71

3.17. PROGRAM LIST 'STANDS' COMMAND SAMPLE OUTPUT 3- 73

3.18. PROGRAM LIST 'STANDS AND PLOTS' COMMAND SAMPLE OUTPUT 3- 75

3.19. PROGRAM LIST 'DATA' COMMAND SAMPLE OUTPUT 3- 77

3.20. PROGRAM REVIEW SAMPLE INPUT STREAM 3- 89

3.21. PROGRAM REVIEW SAMPLE TITLE PAGE OUTPUT 3- 91

3.22. PROGRAM REVIEW SAMPLE REVIEW PAGE OUTPUT 3- 92

3.23. PROGRAM REVIEW SAMPLE CLOSING PAGE OUTPUT 3- 93

3.24. PROGRAM SEARCH SAMPLE INPUT STREAM 3-110

3.25. PROGRAM SEARCH INPUT REFERENCE FORM 3-124

3.26. PROGRAM SEARCH SAMPLE TITLE PAGE OUTPUT 3-126

3.27. PROGRAM SEARCH SAMPLE INPUT ECHO PAGE OUTPUT 3-127

3.28. PROGRAM SEARCH SAMPLE TABLE OUTPUT 3-128

3.29. PROGRAM SEARCH SAMPLE TABLE OUTPUT 3-129

3.30. PROGRAM SEARCH SAMPLE TABLE OUTPUT 3-130

3.31. PROGRAM SEARCH SAMPLE PERIPHERAL DATA FILE STRUCTURE OUTPUT 3-131

1. INTRODUCTION

1. INTRODUCTION

1. INTRODUCTION

1.1. PURPOSE AND STRUCTURE

Although considerable fuels inventory data are available for forest and range ecosystems of the western United States, it exists in many diverse formats at widely scattered locations. Needed is a uniform ADP-compatible database available for quantitative efforts to describe, classify, and rate fuel communities. The NATIONAL FUELS INVENTORY LIBRARY (NFIL) was constructed to store, retrieve, and analyze fuels inventory data for such purposes.

The NFIL is a two part system designed to (1) bring together all fuels inventory data in its many diverse formats into a common database which (2) may be accessed by NFIL analysis routines or other standardized statistical packages. The corresponding functional parts of the NFIL are:

- Fuels Inventory Database (FID)
- Management Software Package (MSP)

The Fuels Inventory Database consists of three permanent files stored on magnetic tape or other mass storage device. The files store and document all fuels inventory information in the NFIL. The three files are:

- Inventory Data File
- Stand Documentation File
- Variable Index File

The FID files are introduced briefly in Section 1.2. and described

in detail in Section 2.

The Management Software Package consists of six ASCII standard FORTRAN IV computer programs. The programs build and maintain the FID files and retrieve and summarize fuels information from the Library.

Three MSP programs perform FID file construction and maintenance functions:

- PROGRAM INDEX
- PROGRAM DATA
- PROGRAM COUNT

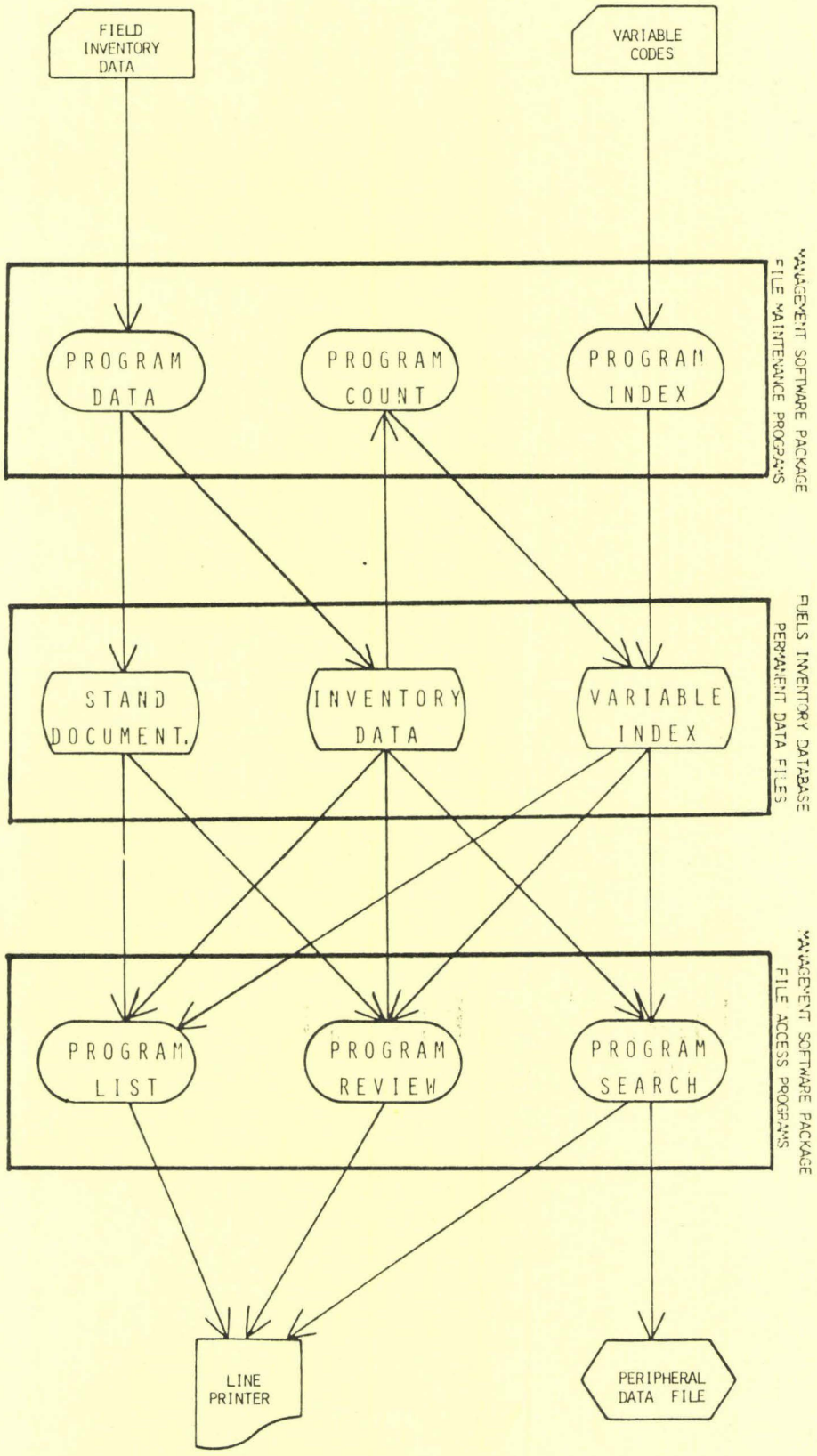
The remaining three MSP programs retrieve, list, and summarize selected fuels inventory information:

- PROGRAM LIST
- PROGRAM REVIEW
- PROGRAM SEARCH

The MSP programs are introduced briefly in Section 1.3. and described in detail in Section 3.

The functional relationships between the Fuels Inventory Database files and the Management Software Package programs are illustrated in Figure 1.1 .

FIGURE 1.1 : FUNCTIONAL RELATIONSHIPS BETWEEN THE FUELS INVENTORY DATABASE FILES AND MANAGEMENT SOFTWARE PACKAGE PROGRAMS



1.2. INTRODUCTION TO THE FUELS INVENTORY DATABASE

1.2 NFIL FUELS INVENTORY DATABASE FILES (FID)

The NFIL Fuels Inventory Database consists of three independent permanent files:

- Inventory Data File (Sections 1.2.1. and 2.1.)
- Stand Documentation File (Sections 1.2.2. and 2.2.)
- Variable Index File (Sections 1.2.3. and 2.3.)

A temporary data file is also produced as output from MSP PROGRAM SEARCH:

- Peripheral Data File (Sections 2.4. and 3.6.)

1.2.1. Inventory Data File

The Inventory Data File is the heart of the NFIL, containing all the fuels inventory data.

The inventory data are stored as 'code:value' pairs with one pair for each piece of inventory information on file. The 'code' is a 4 byte integer identifying the inventory variable whose 'value' is stored in the following 8 byte integer field. A unique code is assigned by the Library user to every new location, stand, site, and fuel variable type to appear in the FID. New codes must be entered and documented in the Variable Index File before they will be accepted by the Inventory Data File.

The 'value' is an 8 byte integer field containing the inventory variable's measure (if continuous), count (if discrete), or class code (if ordinal). Continuous variable values are converted from integer to floating point decimal using a signed 4 byte base 10 exponent stored in the Variable Index File. To illustrate, a continuous inventory variable with a base 10 conversion factor of -3 and a value field containing 12345678 would have its value converted to 12345.678 during MSP program retrieval and analysis. The base 10 exponent for count and ordinal variables is 0.

The code:value pairs for each inventory entry are stored contiguously on the Inventory Data File. An 'inventory entry' is the smallest level of inventory information whose code:value pairs are specific to a given location, stand, and site. An inventory entry may represent a single inventory 'plot', or it may contain summary values for several plots. Regardless, the entry is defined by a single appearance of each variable code recorded, e.g., there is only 1 cover type, 1 elevation, and/or 1 total fuel load per entry. Plot level inventory entries are generally preferrable over stand summary entries since they are more highly resolved and site specific. The MSP programs are designed to provide stand summary statistics for the Library user, and stand summary entries are therefore unnecessary.

Each inventory entry's block of contiguous code:value pairs is headed by a key:size pair. The 'key' is an 8 byte integer sequentially assigned to inventory entries as they are added to the Inventory Data File. It serves the dual purposes of a unique identification code for the entry and as an address to the entry's location in the Inventory Data File. The 'size' is a 4 byte integer denoting the number of trailing code:value pairs for the entry.

The code:value pair concept has several advantages over a preformatted data file:

- a preformatted file requires one field reserved for every existing or potential variable type to be included in the database

- original data integrity is maintained -- recoding is not required to fit new data into the existing format, and continuous data need not be assigned to discrete classes
- new inventory variables may be introduced at any time without restructuring data files or changing applications software
- mass storage requirements are reduced

The Inventory Data File is documented in Section 2.1.

1.2.2. Stand Documentation File

The Stand Documentation File is a catalog of inventory 'stands' and entries residing on the Inventory Data File. It is provided mainly for Library user reference to the amount and type of inventory information available in the NFIL.

The Library user identifies groups of inventory entries as members of a single 'stand' at the time new inventory data is added to the FID. A 'stand' may be defined to suit the user's purposes; it may be a collection of inventory plot entries from a single forest overstory/understory community, or a set of inventory summary entries from a single geographic location or administrative unit. The user includes any inventory documentation and stand and site descriptions he wishes to appear in the Stand Documentation File for each stand identified.

For every user-identified stand of inventory information, the Stand Documentation File contains the Library-assigned stand number, the user-supplied stand description and documentation, and a list of each inventory entry for the stand. The inventory entry list includes each entry's original 8 byte alphameric identifier, the new Library-assigned inventory entry 'key', and the number of inventory plots represented by the entry (equals 1 for individual plot entries, and is greater than 1 for summary entries).

The Stand Documentation File is documented in Section 2.2.

1.2.3. Variable Index File

The Variable Index File documents all inventory variable codes and units of measure used in the FID. The file records each inventory variable's 4 byte integer code, a 56 byte alphameric description, the variable's signed 4 byte integer base 10 conversion factor, the number of inventory entries in the Inventory Data File referencing the variable, and a list of the variable's classes.

In the case of continuous and count variables, the class list has a single line entry indicating the variable's units of measure. In the case of ordinal variables, the list has 1 line per class containing a 4 byte integer class code, a 56 byte alphameric class description, and the number of inventory entries in the Inventory Data File referencing the particular variable class.

The Variable Index File is accessed by both the Library user and MSP programs to identify inventory variables. The file therefore serves as a catalog to variable codes required as input to the MSP programs. When adding new inventory information to the FID, the user must identify each piece of information as a code:value pair. New variable types must have their codes entered into the Variable Index File before their code:value pairs will be accepted by the Inventory Data File.

As noted above, the Variable Index File includes the number of references in the Inventory Data File to each variable and variable class. This feature enables the Library user to quickly review the type and amount of inventory information currently in the FID. For example, the Variable Index File may show 10,000 inventory entries in the Inventory Data File have a forest cover type variable included, of which 4000 are classed as Douglas-fir, 4000 as lodgepole pine, and 2000 as other forest cover types.

The Variable Index File is documented in Section 2.3.

1.3. INTRODUCTION TO THE MANAGEMENT SOFTWARE PACKAGE

1.3. NFIL MANAGEMENT SOFTWARE PACKAGE (MSP)

The Management Software Package is the NFIL librarian. It consists of six ASCII standard FORTRAN IV computer programs. Three programs are used to create, build, and maintain the Fuels Inventory Database:

- PROGRAM INDEX (Sections 1.3.1. and 3.1.)
- PROGRAM DATA (Sections 1.3.2. and 3.2.)
- PROGRAM COUNT (Sections 1.3.3. and 3.3.)

The remaining three MSP programs are used to retrieve, list, and summarize selected fuels inventory data:

- PROGRAM LIST (Sections 1.3.4. and 3.4.)
- PROGRAM REVIEW (Sections 1.3.5. and 3.5.)
- PROGRAM SEARCH (Sections 1.3.6. and 3.6.)

1.3.1. PROGRAM INDEX

1.3.1.1. Purpose

PROGRAM INDEX is an NFIL file maintenance routine. It is used to create a new Variable Index File or to add new variables or variable classes to an existing Variable Index File.

1.3.1.2. Input

The first input card image instructs PROGRAM INDEX to either CREATE a new Variable Index File or to ADD new variables or variable classes to an existing file. It is followed by one card input image for each new variable or variable class to be entered. The card image includes the 4 byte integer variable code, 4 byte integer variable class code, signed 4 byte integer base 10 exponent conversion factor, and a 56 byte alphameric variable or variable class description.

1.3.1.3. Output

A single line of output is written for each input card image indicating whether it was accepted or rejected by PROGRAM INDEX. If accepted, the message will state whether a new variable or a new variable class was created. If rejected, an error message will appear explaining the reason. A closing message is also printed stating the number of input card images read and whether or not they were permanently added to the Variable Index File.

PROGRAM INDEX is fully documented in Section 3.1.

1.3.2. PROGRAM DATA

1.3.2.1. Purpose

PROGRAM DATA is an NFIL file maintenance routine. It is used to create a new Inventory Data File and Stand Documentation File, or to add new inventory data and documentation to existing files.

1.3.2.2. Input

The user arranges groups of inventory entries into 'stands' using a NEW STAND input command for each inventory entry group. The NEW STAND card image is followed by stand documentation images, then by sets of code:value pairs, one set per inventory entry to be included in the stand. Any number of NEW STAND cards may be included in the input stream, and any number of inventory entries may be included in a stand. Every stand, however, must have at least 1 inventory entry, and every inventory entry must be assigned to a stand, even if it is the only stand entry.

A stand may be defined to suit the Library user's purposes; it may be a collection of inventory plot entries from a single forest community, or a set of inventory summary entries from a single geographic location or administrative unit.

Any number of stand documentation cards may appear after the NEW STAND card image, each card containing 76 bytes of alphameric information. The stand documentation will appear verbatum in the Stand Documentation File, and complete stand documentation is the Library user's responsibility.

One or more sets of input images follows the stand documentation cards, one set for each inventory entry to be included in the stand. The first card image of the set contains an 8 byte alphameric inventory entry identification (usually the field plot number, to be entered onto the Stand Documentation File for cross-reference back to the original field forms), the number of plots represented by the entry (equals 1 if inventory plot entries, greater than 1 if inventory plot summaries), and the number of trailing code:value images for the entry. The entry's code:value pairs follow, one pair per input card image.

1.3.2.3. Output

The Library sequentially assigns an 8 byte integer stand number to each new stand as it is added to the Fuels Inventory Database. An 8 byte integer 'key' is also sequentially assigned to every inventory entry. Stand numbers and inventory entry keys are independent of one another.

Each NEW STAND card image and its trailing documentation is echoed back to the lineprinter. It is followed by a single line of output giving the new stand's number, number of inventory entries for the stand, and keys for the first and last stand inventory entries.

The new stand number, documentation, number of inventory entries, and a list of each inventory entry's key, 8 byte alphameric description, and number of inventory plots represented by the entry are written to the Stand Documentation File.

Each inventory entry's 8 byte integer key and number of trailing code:value pairs are entered into the Inventory Data File, followed by its code:value pairs. Three new code:value pairs are created for each inventory entry by PROGRAM DATA; one pair indicating the entry key, one pair indicating the entry's stand number, and a third pair indicating the number of plots represented by the entry.

PROGRAM DATA is fully documented in Section 3.2.

1.3.3. PROGRAM COUNT

1.3.3.1. Purpose

PROGRAM COUNT is an NFIL file maintenance routine. It counts the occurrences of every variable code and variable class code in the Inventory Data File and enters the totals into the Variable Index File.

1.3.3.2. Input

PROGRAM COUNT does not read card image input.

1.3.3.3. Output

PROGRAM COUNT produces no lineprinter output. The number of occurrences of each variable and variable class code are entered into the Variable Index File as an 8 byte integer.

PROGRAM COUNT is fully documented in Section 3.3.

1.3.4. PROGRAM LIST

1.3.4.1. Purpose

PROGRAM LIST is an NFIL data retrieval routine. It is used to produce complete or partial listings of the Variable Index File, Stand Documentation File, and/or Inventory Data File.

1.3.4.2. Input

The file to be listed is indicated by a card image containing the file command INDEX, STAND, STAND AND PLOT, or DATA. It is followed by a specification list containing the variable codes (from the Variable Index File), stands or inventory entries (from the Stand Documentation File), or inventory entries (from the Inventory Data File) to be listed. Any number of file list commands may appear in a single PROGRAM LIST run.

The specification list is a series of card images containing two 8 byte integer fields. The integers correspond to the first and last contiguous variable codes, stand numbers, or inventory entry keys to be listed. Any number of specification pairs may follow a single file command. If no specification list follows the file command, the entire file is listed.

1.3.4.3. Output

PROGRAM LIST outputs the requested variable and variable class records from the Variable Index File, requested stand or stand and inventory entry lists from the Stand Documentation File, and/or requested inventory entries from the Inventory Data File. All output is reformatted to increase readability and reduce volume.

PROGRAM LIST is fully documented in Section 3.4.

1.3.5. PROGRAM REVIEW

1.3.5.1. Purpose

PROGRAM REVIEW is an NFIL data retrieval and summary routine. It permits Library users to review information amount and content of selected inventory data sets. Any set of inventory entries may be reviewed as a unit, whether members of a single stand, groups of stands, or groups of inventory entries from various stands. The program reviews the occurrence of each variable and variable class code within the selected data set and provides descriptive statistics of continuous variable values.

1.3.5.2. Input

A card image containing the command STANDS or PLOTS indicates whether the review set members are to be identified by stand number or by inventory entry key. The command is followed by a specification list indicating the stands or inventory entries to be included in the review set.

The specification list consists of any number of card images, each containing two 8 byte integer fields. The fields contain the first and last contiguous stand numbers or inventory entry keys to be included in the review. All stands or inventory entries appearing explicitly or implicitly in the list will be reviewed as a single unit until the next STANDS or PLOTS command is read.

1.3.5.3. Output

A complete set of output is produced for every appearance of a STANDS or PLOTS command. For every variable appearing at least once in the reviewed data set, its code, alphameric description, absolute occurrence, and percent occurrence (relative to the total number of inventory entries reviewed) are printed. For ordinal variables, a line is printed for every class of the variable including the class code, alphameric description, absolute occurrence, and percent occurrence (relative to the variable's absolute occurrence). For continuous and count variables, the mean, standard deviation, minimum and maximum values, and units of measure are printed.

PROGRAM REVIEW is fully documented in Section 3.5.

1.3.6. PROGRAM SEARCH

1.3.6.1. Purpose

PROGRAM SEARCH is an NFIL data retrieval, analysis, and summary routine. It permits the Library user to:

- search the Fuels Inventory Database and retrieve values of selected inventory variables (for example, the user may wish to retrieve only those inventory entries which have total fuel load, cover type, stand age, and national forest region recorded)
- constrain retrieved entries to those with variables whose values are within selected bounds (the user may further wish to restrict retrieved entries to those in Douglas-fir, lodgepole pine, or ponderosa pine cover types between 3000 and 6000 feet elevation from Region One national forests)
- stratify retrieved inventory entries into one, two, or no stratifications, and define classes within each (a cover type stratification of three classes by four stand age classes)
- define variables as constraint, stratification, and/or table variables
- create a temporary peripheral data file containing table variable values and their stratification class assignments
- perform statistical analyses of retrieved table variable values within the selected stratification scheme, including:
  - -- absolute occurrence
  - -- percent occurrence
  - -- sum
  - -- sum of squares
  - -- mean
  - -- variance
  - -- standard deviation
  - -- standard error
  - -- percent presence
  - -- minimum value
  - -- maximum value

1.3.6.2. Input

PROGRAM SEARCH uses 15 input commands to perform the above functions. The command syntax is patterned after other standardized statistical analysis packages. Section 3.6.2. discusses PROGRAM SEARCH input commands, syntax, and formats in full detail.

1.3.6.3. Output

PROGRAM SEARCH echoes the card image input stream back to the line-printer as the first page of output for each requested search. Error messages will appear after any input image using incorrect syntax or exceeding program limitations.

One table is produced for every table variable-statistic requested. For example, if four table variables are declared, each with an occurrence, mean, and standard deviation estimate, 12 tables will be printed. Each table includes a caption indicating the run number and name, table variable code and name, and type of analysis statistic (occurrence, mean, standard deviation, etc.). Row and column headings for stratified analyses appear as specified in the input stream. Tables may be one or two pages, depending on the number of stratification classes defined.

PROGRAM SEARCH is fully documented in Section 3.6.

F

I

D

2. FUELS INVENTORY DATABASE

2.1. INVENTORY DATA FILE

2.1. INVENTORY DATA FILE

2.1.1. Purpose

The Inventory Data File contains all inventory entry variable codes and variable values.

2.1.2. Structure

The Inventory Data File consists of 4 record types:

- I File Label (80 alphameric bytes)
- II File Size (8 byte integer)
- III Inventory Entry Key:Size Pair (12 byte integer)
- IV Inventory Entry Code:Value Pair (12 byte integer)

Each record type is documented in the following pages and examples of each are illustrated in Figure 2.2.

2.1.2.1. Inventory Data File Record Type I - File Label

FILE NAME Inventory Data File

RECORD NAME File Label

RECORD TYPE I OF IV

RECORD LENGTH 80 bytes

DESCRIPTION Identifies the file as the Inventory Data File, and gives the date PROGRAM COUNT last determined the file's variable and variable class code occurrences. It is the first record of the file, and appears only once.

COLUMNS	FORMAT	VARIABLE NAME	DESCRIPTION
1 - 68	17A4	LABEL(17)	Alphameric file label
69 - 80	3A4	LABEL(20)	Alphameric date of last PROGRAM COUNT file read

2.1.2.2. Inventory Data File Record Type II - File Size

FILE NAME Inventory Data File

RECORD NAME File Size

RECORD TYPE II OF IV

RECORD LENGTH 8 bytes

DESCRIPTION Records the number of inventory entries residing on the file. It is the second record of the file, and appears only once.

COLUMNS	FORMAT	VARIABLE NAME	DESCRIPTION
1 - 8	I8	NKEYS	Number of fuel inventory entries on the data file

2.1.2.3. Inventory Data File Record Type III - Key:Size Pair

FILE NAME Inventory Data File

RECORD NAME Inventory Entry Key:Size Pair

RECORD TYPE III OF IV

RECORD LENGTH 12 bytes

DESCRIPTION Contains the inventory entry's 8 byte integer key and 4 byte integer indicating the number of trailing code: value records for the entry. The record appears once as the first record of every inventory entry.

COLUMNS	FORMAT	VARIABLE NAME	DESCRIPTION
1 - 8	I8	KEY	Inventory entry primary key
9 - 12	I4	NPAIRS	Number of trailing code:value pairs

2.1.2.4. Inventory Data File Record Type IV - Code:Value Pair

FILE NAME Inventory Data File

RECORD NAME Inventory Entry Code:Value Pair

RECORD TYPE IV OF IV

RECORD LENGTH 12 bytes

DESCRIPTION Contains the variable's 4 byte code and 8 byte value. Appears once for every inventory variable recorded for the inventory entry.

COLUMNS	FORMAT	VARIABLE NAME	DESCRIPTION
1 - 4	I4	KODE	User-assigned variable code
5 - 12	I8	KVALUE	Variable value, 8 significant digits with no decimal point

2.1.3. Maintenance and Use

2.1.3.1. Write Access

The Inventory Data File is built and maintained by MSP PROGRAM DATA. PROGRAM COUNT also write-accesses the Inventory Data File to update the File Label (record type I) date. The date indicates the last time PROGRAM COUNT was used to determine the number of occurrences of every variable and variable class code in the file.

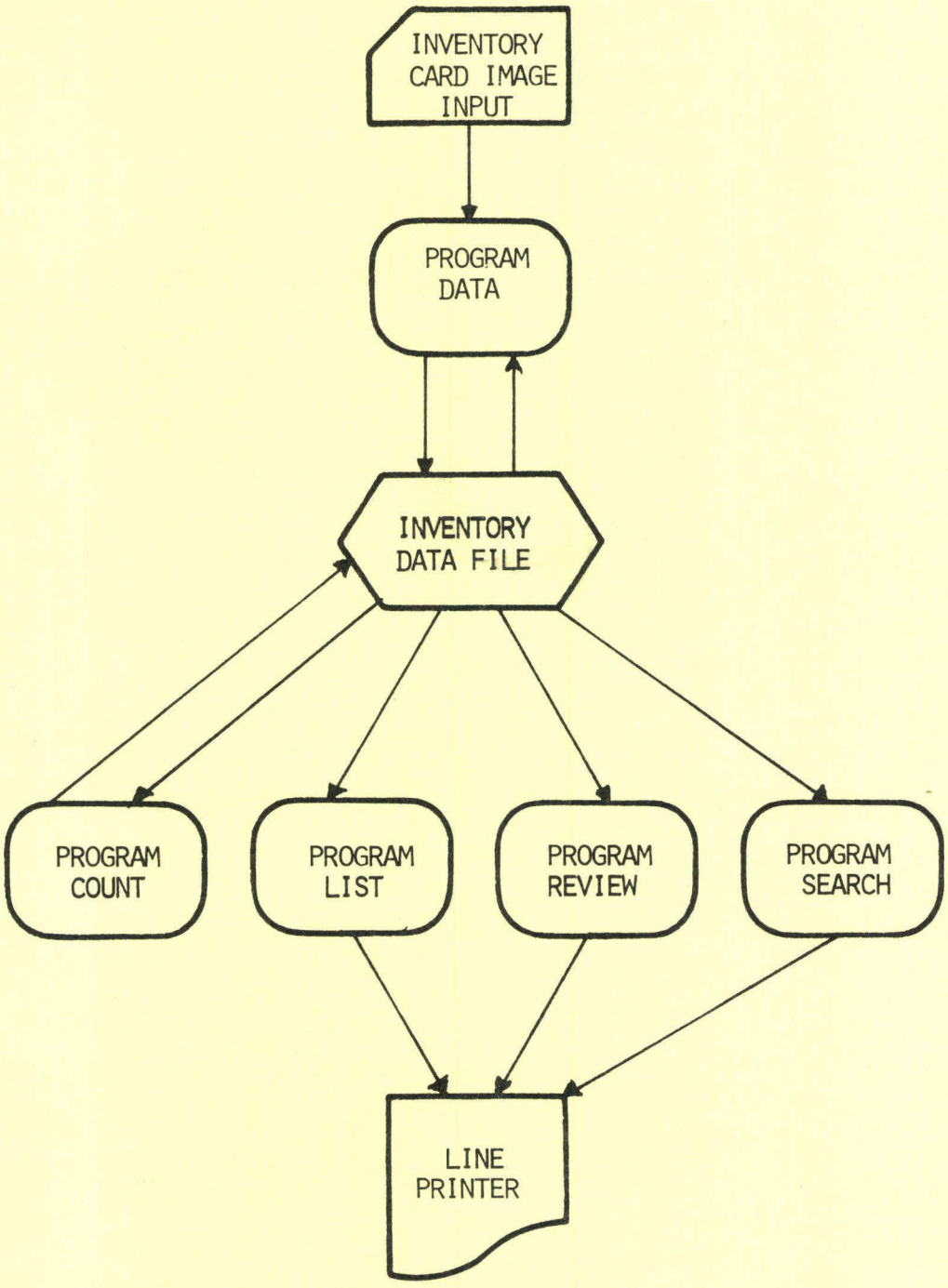
2.1.3.2. Read Access

The Inventory Data File is read-accessed by the following MSP programs:

- PROGRAM DATA
- PROGRAM COUNT
- PROGRAM LIST
- PROGRAM REVIEW
- PROGRAM SEARCH

Figure 2.1. illustrates MSP program read- and write-access to the Inventory Data File.

FIGURE 2.1. INVENTORY DATA FILE ACCESS



2.1.4. Inventory Data File Sample Listing

A sample listing of the Inventory Data File appears in Figure 2.2. Record types I, II, and III are indicated in the left margin. All unmarked records are of record type IV.

A complete listing of the Inventory Data File in condensed format may be obtained using the PROGRAM LIST input card image

```
1234567890
DATA
```

Figure 2.3. illustrates PROGRAM LIST output for the same sample Inventory Data File in Figure 2.2.

FIGURE 2.2. INVENTORY DATA FILE SAMPLE LISTING

RECORD TYPE I
RECORD TYPE II
RECORD TYPE III

```
NFIL  -  INVENTORY DATA FILE  -  CREATED  22 SEP 79
      60
           1  18
     1        1
     2        1
     3        1
   101        3
   111       25
   201      202
   211       25
   221        2
   301     3780
   411      449
   421     2457
   431   133544
   441   136550
   501       12
   601    13349
   701        1
   702    11119
   801   147579
```

RECORD TYPE III

```
           2  18
     1        2
     2        1
     3        1
   101        3
   111       26
   201      202
   211       25
   221        2
   301     3954
   411      536
   421     3079
   431    95124
   441    98739
   501       14
   601    14227
   701        1
   702    18614
   801   117353
```

RECORD TYPE III

```
           3  18
     1        3
     2        1
     3        1
   101        3
   111       26
   201      202
   211       25
   221        2
   301     3437
   411      439
   421      355
   431    58071
   441    59425
   501        9
   601     8289
   701        1
   702     9124
   801    68549
```

RECORD TYPE III

```
           4  18
     1        4
     2        1
```

FIGURE 2.3. INVENTORY DATA FILE SAMPLE LISTING

PRODUCED BY PROGRAM LIST 'DATA' COMMAND

```
FIL - INVENTORY DATA FILE - CREATED 22 SEP 79. LAST COUNTED ON

NUMBER OF FUEL INVENTORIES CURRENTLY ON THE DATA FILE IS      100

  1    18 VARIABLES
         1         1         2         1         3         1       101         3       111        25       201       202
       211        25       221         2       301      3780       411       449       421      2467       431    133644
       441    136560       501        12       601     13389       701         1       702     11119       801    147679

  2    18 VARIABLES
         1         2         2         1         3         1       101         3       111        25       201       202
       211        25       221         2       301      3984       411       535       421      3079       431     95124
       441     98734       501        14       601     14227       701         1       702     18614       801    117353

  3    18 VARIABLES
         1         3         2         1         3         1       101         3       111        25       201       202
       211        25       221         2       301      3437       411       489       421       865       431     58071
       441     59425       501         9       601      8287       701         1       702      9124       801     68549

  4    18 VARIABLES
         1         4         2         1         3         1       101         3       111        25       201       202
       211        25       221         2       301      4112       411       399       421      3048       431    100635
       441    104082       501         5       601     31654       701         1       702      6085       801    110167

  5    18 VARIABLES
         1         5         2         1         3         1       101         3       111        25       201       202
       211        25       221         2       301      3272       411       504       421      2285       431     79962
       441     82451       501        15       601      9948       701         1       702      6836       801     89687

  6    18 VARIABLES
         1         6         2         1         3         1       101         3       111        25       201       202
       211        25       221         2       301      4044       411       600       421      1747       431    135329
       441    137575       501         5       601     25663       701         1       702      5481       801    144157

  7    18 VARIABLES
         1         7         2         1         3         1       101         3       111        25       201       202
       211        25       221         2       301      3941       411       325       421      1529       431     99873
       441    101725       501         9       601     11355       701         1       702     13136       801    114864

  8    18 VARIABLES
         1         8         2         1         3         1       101         3       111        25       201       202
       211        25       221         2       301      3245       411       540       421      2097       431    170558
       441    173195       501        10       601     14529       701         1       702     13093       801    185293
```

2.2. STAND DOCUMENTION FILE

2.2. STAND DOCUMENTATION FILE

2.2.1. Purpose

The Stand Documentation File is a catalog of inventory stands and entries residing on the Inventory Data File. It is provided mainly for Library user reference to the amount and type of inventory information available on the NFIL.

2.2.2. Structure

The Stand Documentation File consists of 5 record types:

- I File Label (80 alphameric btyes)
- II File Size (8 byte integer)
- III Stand Identification (28 byte integer)
- IV Stand Description (76 alphameric bytes)
- V Inventory Entry Identification (20 alphameric bytes)

Each record type is documented in the following pages and examples of each are illustrated in Figure 2.5.

2.2.2.1. Stand Documentation File Record Type I - File Label

FILE NAME Stand Documentation File

RECORD NAME File Label

RECORD TYPE I OF V

RECORD LENGTH 80 BYTES

DESCRIPTION Identifies the file as the Stand Documentation File. It is the first record of the file and appears only once.

COLUMNS	FORMAT	VARIABLE NAME	DESCRIPTION
1 - 80	20A4	LABEL(20)	Alphameric file label

2.2.2.2. Stand Documentation File Record Type II - File Size

FILE NAME Stand Documentation File

RECORD NAME File Size

RECORD TYPE II OF V

RECORD LENGTH 8 BYTES

DESCRIPTION Records the number of stands documented in the file. It is the second record of the file and appears only once.

COLUMNS	FORMAT	VARIABLE NAME	DESCRIPTION
1 - 8	I8	NSTANDS	Number of stands on the file

2.2.2.3. Stand Documentation File Record Type III - Stand Identification

FILE NAME Stand Documentation File

RECORD NAME Stand Identification

RECORD TYPE III OF V

RECORD LENGTH 28 BYTES

DESCRIPTION Contains the stand's 8 byte integer Library-assigned number, the number of inventory entries for the stand, the key of the stand's first inventory entry on the Inventory Data File, and the number of documentation card images (record type IV) for the stand. The record appears once for every stand documented in the file and is the first record for the stand.

COLUMNS	FORMAT	VARIABLE NAME	DESCRIPTION
1 - 8	I8	IDSTND	Stand number
9 - 16	I8	INVENT	Number of inventory entries for the stand
17 - 24	I8	KEY1	Primary key of the stand's first inventory entry on the Inventory Data File (location on the Inventory Data File at which the stand's INVENT contiguous inventory entries begin)
25 - 28	I4	NDOC	Number of trailing Stand Description records (record type IV)

2.2.2.4. Stand Documentation File Record Type IV - Stand Description

FILE NAME Stand Documentation File

RECORD NAME Stand Description

RECORD TYPE IV OF V

RECORD LENGTH 76 BYTES

DESCRIPTION Contains the user-supplied alphameric stand and site description. Any number of Stand Description records may follow a Stand Identification record.

COLUMNS	FORMAT	VARIABLE NAME	DESCRIPTION
1 - 76	19A4	ADOC(19)	Alphameric stand description

2.2.2.5. Stand Documentation File Record Type V - Inventory Entry Identification

FILE NAME Stand Documentation File

RECORD NAME Inventory Entry Identification

RECORD TYPE V OF V

RECORD LENGTH 20 BYTES

DESCRIPTION Contains the inventory entry key, number of inventory plots represented by the entry, and alphameric entry identification. One Inventory Entry Identification record appears for every inventory entry in the stand. The Inventory Entry Identification records follow the Stand Description records for each stand. One inventory identification record appears in the Stand Documentation File for every inventory entry on the Inventory Data File.

COLUMNS	FORMAT	VARIABLE NAME	DESCRIPTION
1 - 8	I8	KEY	Inventory entry key
9 - 12	I4	NPLOTS	Number of inventory plots represented by the entry
12 - 20	2A4	ENTRY(2)	Alphameric entry description (usually corresponds to the entry's original field inventory identification code)

2.2.3. File Maintenance and Use

2.2.3.1. Write Access

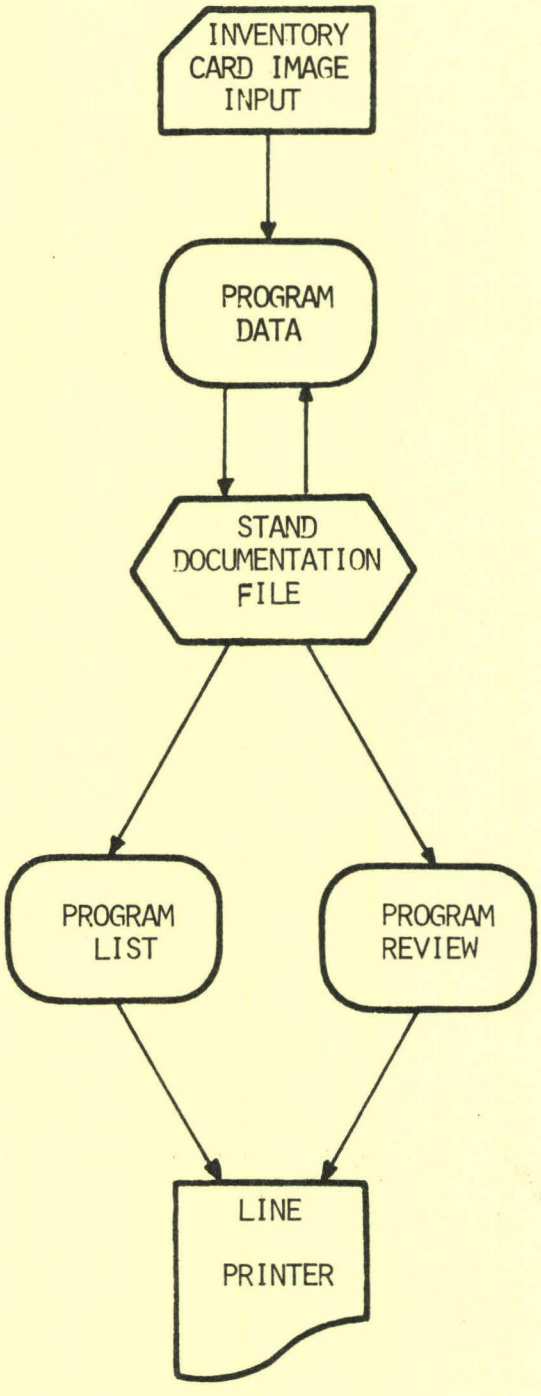
The Stand Documentation File is built and maintained by PROGRAM DATA, which is the only MSP program with write-access to the file.

2.2.3.2. Read Access

The Stand Documentation File is read-accessed by PROGRAMs LIST and REVIEW.

Figure 2.4. illustrates MSP program read- and write-access to the Stand Documentation File.

FIGURE 2.4. STAND DOCUMENTATION FILE ACCESS



2.2.4. Stand Documentation File Sample Listings

A sample listing of the Stand Documentation File appears in Figure 2.5. Record types I, II, II, and IV are indicated in the left margin. All unmarked records are of record type V.

A complete listing of the Stand Documentation File in condensed format may be obtained using the PROGRAM LIST input card image

```
          1
1234567890
STAND
```

or

```
          1         2
12345678901234567890
STAND AND PLOT
```

The first command image will produce a listing of the Stand Documentation record types I, II, II, and IV as in Figure 2.6. The second command image will produce a listing of all 5 record types on the file as in Figure 2.7.

FIGURE 2.5. STAND DOCUMENTATION FILE SAMPLE LISTING

```
REC. TYPE I     NATL FUELS INVENTORY LIBRARY - STAND DOCUMENTATION FILE - CREATED 22 SEP 79
REC. TYPE II          5
                      1      10        1    6
REC. TYPE III   STAND ID - ALPHA
                LOCATION - BITTERROOT NF, SOUTH FORK RD, SW MONTANA, USA
                STAND    - SECOND GROWTH DOUGLAS-FIR AFTER 1940 CLEARCUT
REC. TYPE IV    SITE     - 4700 FT, NE ASPECT, HIGH PRODUCTIVITY
                FUELS    - NATURAL
                DATE     - JULY 1977
                     1   1ALPHA  1
                     2   1ALPHA  2
                     3   1ALPHA  3
                     4   1ALPHA  4
REC. TYPE V          5   1ALPHA  5
                     6   1ALPHA  6
                     7   1ALPHA  7
                     8   1ALPHA  8
                     9   1ALPHA  9
                    10   1ALPHA 10
REC. TYPE III        2      10       11    6
                STAND ID - BETA
                LOCATION - BITTERROOT NF, WEST FORK RD, SW MONTANA, USA
REC. TYPE IV    STAND    - OLD GROWTH DOUGLAS-FIR REPLACING DECADENT LODGEPOLE STAND
                SITE     - 5300 FT, EAST ASPECT
                FUELS    - NATURAL
                DATE     - AUGUST 1977
                    11   1BETA   1
                    12   1BETA   2
                    13   1BETA   3
                    14   1BETA   4
                    15   1BETA   5
REC. TYPE V         16   1BETA   6
                    17   1BETA   7
                    18   1BETA   8
                    19   1BETA   9
                    20   1BETA  10
REC. TYPE III        3      10       21    6
                STAND ID - GAMMA
                LOCATION - BITTERROOT NF, DARBY RD, SW MONTANA, USA
REC. TYPE IV    STAND    - EVEN-AGED LODGEPOLE PINE, 75 YEARS AFTER FIRE
                SITE     - 7400 FT, SOUTH ASPECT
                FUELS    - NATURAL
                DATE     - AUGUST 1977
                    21   1GAMMA  1
                    22   1GAMMA  2
                    23   1GAMMA  3
                    24   1GAMMA  4
                    25   1GAMMA  5
                    26   1GAMMA  6
REC. TYPE V         27   1GAMMA  7
                    28   1GAMMA  8
                    29   1GAMMA  9
                    30   1GAMMA 10
REC. TYPE III        4      10       31    6
                STAND ID - ALPHA
                LOCATION - LOLO NF, SEELEY LAKE RD, WEST-CENTRAL MONTANA, USA
REC. TYPE IV    STAND    - OLD GROWTH DOUGLAS-FIR REPLACING DECADENT LODGEPOLE PINE
                SITE     - 4500 FT, NW ASPECT
                FUELS    - NATURAL
                DATE     - JULY 1978
                    31   1ALPHA  1
                    32   1ALPHA  2
```

FIGURE 2.6. STAND DOCUMENTATION FILE SAMPLE LISTING

PRODUCED BY PROGRAM LIST 'STANDS' COMMAND

```
NATL FUELS INVENTORY LIBRARY - STAND DOCUMENTATION FILE - CREATED 22 SEP 79

NUMBER OF FUEL INVENTORY STANDS CURRENTLY ON FILE IS 10

STAND NO. 1   10 INVENTORY ENTRIES ON FILE (ENTRY NOS. 1 - 10)
STAND ID - ALPHA
LOCATION - BITTERROOT NF, SULA RD, SW MONTANA, USA
STAND - SECOND GROWTH DOUGLAS-FIR AFTER 1940 CLEARCUT
SITE - 3700 FT, NE ASPECT, HIGH PRODUCTIVITY
FUELS - NATURAL
DATE - JULY 1977

STAND NO. 2   10 INVENTORY ENTRIES ON FILE (ENTRY NOS. 11 - 20)
STAND ID - BETA
LOCATION - BITTERROOT NF, WEST FORK RD, SW MONTANA, USA
STAND - OLD GROWTH DOUGLAS-FIR REPLACING DECADENT LODGEPOLE STAND
SITE - 5300 FT, EAST ASPECT
FUELS - NATURAL
DATE - AUGUST 1977

STAND NO. 3   10 INVENTORY ENTRIES ON FILE (ENTRY NOS. 21 - 30)
STAND ID - GAMMA
LOCATION - BITTERROOT NF, DARBY RD, SW MONTANA, USA
STAND - EVEN-AGED LODGEPOLE PINE, 75 YEARS AFTER FIRE
SITE - 7400 FT, SOUTH ASPECT
FUELS - NATURAL
DATE - AUGUST 1977

STAND NO. 4   10 INVENTORY ENTRIES ON FILE (ENTRY NOS. 31 - 40)
STAND ID - ALPHA
LOCATION - LOLO NF, SEELEY LAKE RD, WEST-CENTRAL MONTANA, USA
STAND - OLD GROWTH DOUGLAS-FIR REPLACING DECADENT LODGEPOLE PINE
SITE - 4500 FT, NW ASPECT
FUELS - NATURAL
DATE - JULY 1978

STAND NO. 5   10 INVENTORY ENTRIES ON FILE (ENTRY NOS. 41 - 50)
STAND ID - BETA
LOCATION - LOLO NF, MISSOULA RD, WEST-CENTRAL MONTANA, USA
STAND - LODGEPOLE PINE 80 YEARS AFTER WILDFIRE
SITE - 6500 FT, SW ASPECT
FUELS - NATURAL
DATE - SEPTEMBER 1978

STAND NO. 6   10 INVENTORY ENTRIES ON FILE (ENTRY NOS. 51 - 60)
STAND ID - GAMMA
LOCATION - LOLO NF, SUPERIOR RD, WEST-CENTRAL MONTANA, USA
STAND - LODGEPOLE PINE 125 AFTER WILDFIRE
SITE - 5600 FT, EAST ASPECT
FUELS - NATURAL
DATE - AUGUST 1978
```

FIGURE 2.7. STAND DOCUMENTATION FILE SAMPLE LISTING

PRODUCED BY PROGRAM LIST 'STANDS AND PLOTS' COMMAND

2.3. VARIABLE INDEX FILE

2.3. VARIABLE INDEX FILE

2.3.1. Purpose

The Variable Index File documents all inventory variable codes and units of measure used by the Fuels Inventory Database. It is accessed by both the Library user and MSP programs to identify variable and variable class codes used in the Inventory Data File. The following information about the FID is stored on the Variable Index File:

- variable code
- variable class code
- alphameric variable or variable class description
- variable's base 10 exponent conversion factor
- number of times the variable or variable class code appears in the Inventory Data File

The last item enables the Library user to quickly review the type and amount of inventory information currently residing on the FID.

2.3.2. Structure

The Variable Index File consists of 3 record types:

- I File Label (80 alphameric bytes)
- II File Size (8 byte integer)
- III Variable Documentation (80 alphameric bytes)

Each record type is documented in the following pages and examples of each are illustrated in Figure 2.9.

2.3.2.1. Variable Index File Record Type I - File Label

FILE NAME Variable Index File

RECORD NAME File Label

RECORD TYPE I OF III

RECORD LENGTH 80 BYTES

DESCRIPTION Identifies the file as the Variable Index File. It is the first record of the file and appears only once.

COLUMNS	FORMAT	VARIABLE NAME	DESCRIPTION
1 - 80	20A4	LABEL(20)	Alphameric file label

2.3.2.2. Variable Index File Record Type II - File Size

FILE NAME Variable Index File

RECORD NAME File Size

RECORD TYPE II OF III

RECORD LENGTH 8 BYTES

DESCRIPTION Records the number of variable codes (not variable and variable class codes) stored on the file. Record type II appears only once in the file, and is always the second file record.

COLUMNS	FORMAT	VARIABLE NAME	DESCRIPTION
1 - 8	I8	NCODES	Number of variable codes (excluding variable class codes) on the file

2.3.2.3. Variable Index File Record Type III - Variable Documentation

FILE NAME Variable Index File

RECORD NAME Variable Documentation

RECORD TYPE III OF III

RECORD LENGTH 80 BYTES

DESCRIPTION Contains all the information required by MSP programs to identify variable and variable class codes and to convert continuous variable values to floating point decimals. Record type III is used both for variables and their classes. Only one class exists for continuous and count variables (class code = 9999), and one or more classes may exist for ordinal variables. Record type III appears once for every variable and variable class on the file.

COLUMNS	FORMAT	VARIABLE NAME	DESCRIPTION
1 - 4	I4	KODE1	User-assigned variable code
5 - 8	I4	KLASS	Variable class code (equals 0000 if record is for a variable and not a variable class)
9 - 12	I4	NCLASS	If the record is for a variable, indicates the number of variable classes. If the record is for a variable class, indicates the class's number
13 - 16	I4	MFACTR	signed base 10 exponent variable value conversion factor
17 - 24	I8	NRECS	number of times the variable or variable class code appears in the Inventory Data File
25 - 80	14A4	ACODES(14)	alphameric description of the variable or variable class

2.3.3. File Maintenance and Use

2.3.3.1. Write Access

The Variable Index File is built and maintained by PROGRAM INDEX and PROGRAM COUNT. PROGRAM INDEX is used to create a new file or to add new variables and variable classes to an existing file. PROGRAM COUNT determines the occurrence of every variable and variable class code in the Inventory Data File and enters the count into the 5th field of record type III.

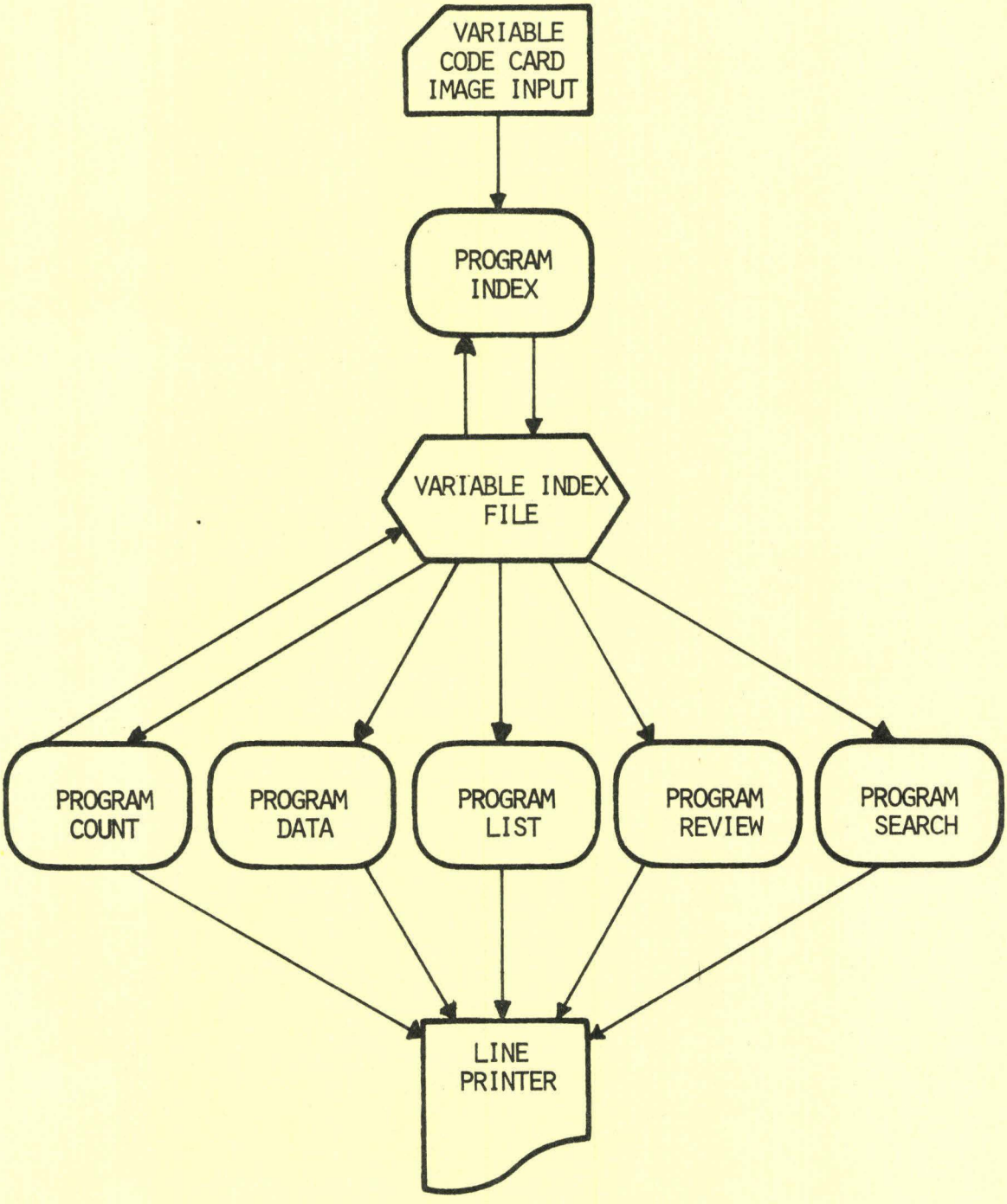
2.3.3.2. Read Access

The Variable Index File is read-accessed by the following MSP programs:

- PROGRAM INDEX
- PROGRAM DATA
- PROGRAM COUNT
- PROGRAM LIST
- PROGRAM REVIEW
- PROGRAM SEARCH

Figure 2.8. illustrates MSP program read- and write-access to the Variable Index File.

FIGURE 2.8. VARIABLE INDEX FILE ACCESS



2.2.4. Variable Index File Sample Listings

A sample listing of the Variable Index File appears in Figure 2.9. Record types I, II, and III are indicated in the left margin.

A complete listing of the Variable Index File in condensed, more readable format may be obtained using the PROGRAM LIST input card image

1234567890
INDEX

Figure 2.10. illustrates PROGRAM LIST output for the same sample Variable Index File in Figure 2.9.

FIGURE 2.9. VARIABLE INDEX FILE SAMPLE LISTING

R.T. I
R.T. II
RECORD TYPE III

```
NATL FUELS INVENTORY LIBRARY -- VARIABLE INDEX FILE --    CREATED   22 SEP 79
      22
    1    0   1   0     100NFIL INVENTORY ENTRY KEY
    19999   1   0     100SEQUENTIALLY-ASSIGNED 8 DIGIT INTEGER
    2    0   1   0     100NFIL STAND NUMBER
    29999   1   0     100SEQUENTIALLY-ASSIGNED 8 DIGIT INTEGER
    3    0   1   0     100NO. OF PLOTS REPRESENTED BY INVENTORY ENTRY
    39999   1   0     1004 DIGIT INTEGER
  101    0   3   0     100USDA NATIONAL FOREST
  101    3   1   0      30BITTERROOT NATIONAL FOREST
  101    5   3   0      40CLEARWATER NATIONAL FOREST
  101   16   2   0      30OLD NATIONAL FOREST
  111    0   2   0     100STATE
  111   12   2   0      40IDAHO
  111   26   1   0      60MONTANA
  201    0   3   0     100FOREST COVER TYPE (BY PLURALITY OF SPECIES)
  201  108   1   0      40_DOGPOLE PINE
  201  202   2   0      50DOUGLAS-FIR
  201  242   3   0      10#ESTERN RED CEDAR
  211    0   1   0     100STAND AGE
  211 9999   1   0     100YEARS
  221    0   3   0     100STAND MGMT STATUS (FUEL TREATMENT)
  221    1   1   0      80OLD GROWTH STAND
  221    2   2   0      20SECOND GROWTH STOCKED STAND
  221   11   3   0       0CLEARCUT LOGGING SLASH
  301    0   1   0     100ELEVATION
  301 9999   1   0     100FEET ABOVE MEAN SEA LEVEL
  311    0   5   0      40ASPECT
  311    1   1   0      10NORTH
  311    3   2   0      10EAST
  311    5   3   0      10SOUTH
  311    7   4   0      10#EST
  311    9   5   0       0FLAT
  411    0   1  -3     1000.25 - 1.0 INCH DEAD + DOWN FUEL LOAD
  411 9999   1  -3     100TONS/ACRE
  421    0   1  -3     1001.0 - 3.0 INCH DEAD AND DOWN FUEL LOAD
  421 9999   1  -3     100TONS/ACRE
  431    0   1  -3     100> 3.0 INCH DEAD AND DOWN FUEL LOAD
  431 9999   1  -3     100TONS/ACRE
  441    0   1  -3     100TOTAL > 0.25 INCH DEAD AND DOWN FUEL LOAD
  441 9999   1  -3     100TONS/ACRE
  501    0   1  -1     100HEIGHEST INTERCEPT < 3 INCH DEAD AND DOWN FUEL
  501 9999   1  -1     100INCHES
  601    0   1  -4     100DEAD AND DOWN FUEL BULK DENSITY (< 3 INCH PARTICLES)
  601 9999   1  -4     100_BS/CU FT
  701    0   2   0     100DUFF FUEL LOAD DETERMINATION METHOD
  701    1   1   0      60DESTRUCTIVE SAMPLE, OVENDRY AND WEIGH
  701   11   2   0      40REGRESSION ON DUFF DEPTH MEASUREMENT
  702    0   1  -3     100DUFF FUEL LOAD
  702 9999   1  -3     100TONS/ACRE
  711    0   1  -1      40DUFF DEPTH
  711 9999   1  -1      40INCHES
  721    0   1  -4      40DUFF FUEL BULK DENSITY
  721 9999   1  -4      40_BS/CU FT
  801    0   1  -3     100TOTAL FUEL LOAD
  801 9999   1  -3     100TONS/ACRE
  901    0   1  -2      40SHRUB COVER
  901 9999   1  -2      40FRACTION OF GROUND AREA COVERED
```

FIGURE 2.10. VARIABLE INDEX FILE SAMPLE LISTING

PRODUCED BY PROGRAM INDEX 'INDEX' COMMAND

NATL FUELS INVENTORY LIBRARY -- VARIABLE INDEX FILE -- CREATED 22 SEP 79

NUMBER OF FUEL INVENTORY VARIABLE CODES NOW ON THE INDEX FILE IS 22

1	NFIL INVENTORY ENTRY KEY	1 CLASSES	1.0E 0 MULT. FACTOR	100 INVENTORY ENTRIES
	9999 SEQUENTIALLY-ASSIGNED 8 DIGIT INTEGER	CLASS NO. 1		100 INVENTORY ENTRIES
2	NFIL STAND NUMBER	1 CLASSES	1.0E 0 MULT. FACTOR	100 INVENTORY ENTRIES
	9999 SEQUENTIALLY-ASSIGNED 8 DIGIT INTEGER	CLASS NO. 1		100 INVENTORY ENTRIES
3	NO. OF PLOTS REPRESENTED BY INVENTORY ENTRY	1 CLASSES	1.0E 0 MULT. FACTOR	100 INVENTORY ENTRIES
	9999 4 DIGIT INTEGER	CLASS NO. 1		100 INVENTORY ENTRIES
101	USDA NATIONAL FOREST	3 CLASSES	1.0E 0 MULT. FACTOR	100 INVENTORY ENTRIES
	3 BITTERROOT NATIONAL FOREST	CLASS NO. 1		30 INVENTORY ENTRIES
	5 CLEARWATER NATIONAL FOREST	CLASS NO. 3		40 INVENTORY ENTRIES
	15 LOLO NATIONAL FOREST	CLASS NO. 2		30 INVENTORY ENTRIES
111	STATE	2 CLASSES	1.0E 0 MULT. FACTOR	100 INVENTORY ENTRIES
	12 IDAHO	CLASS NO. 2		40 INVENTORY ENTRIES
	25 MONTANA	CLASS NO. 1		60 INVENTORY ENTRIES
201	FOREST COVER TYPE (BY PLURALITY OF SPECIES)	3 CLASSES	1.0E 0 MULT. FACTOR	100 INVENTORY ENTRIES
	104 LODGEPOLE PINE	CLASS NO. 1		40 INVENTORY ENTRIES
	202 DOUGLAS-FIR	CLASS NO. 2		50 INVENTORY ENTRIES
	242 WESTERN RED CEDAR	CLASS NO. 3		10 INVENTORY ENTRIES
211	STAND AGE	1 CLASSES	1.0E 0 MULT. FACTOR	100 INVENTORY ENTRIES
	9999 YEARS	CLASS NO. 1		100 INVENTORY ENTRIES
221	STAND MGMT STATUS (FUEL TREATMENT)	3 CLASSES	1.0E 0 MULT. FACTOR	100 INVENTORY ENTRIES
	1 OLD GROWTH STAND	CLASS NO. 1		80 INVENTORY ENTRIES
	2 SECOND GROWTH STOCKED STAND	CLASS NO. 2		20 INVENTORY ENTRIES
	11 CLEARCUT LOGGING SLASH	CLASS NO. 3		0 INVENTORY ENTRIES
301	ELEVATION	1 CLASSES	1.0E 0 MULT. FACTOR	100 INVENTORY ENTRIES
	9999 FEET ABOVE MEAN SEA LEVEL	CLASS NO. 1		100 INVENTORY ENTRIES
311	ASPECT	5 CLASSES	1.0E 0 MULT. FACTOR	40 INVENTORY ENTRIES
	1 NORTH	CLASS NO. 1		10 INVENTORY ENTRIES
	3 EAST	CLASS NO. 2		10 INVENTORY ENTRIES
	5 SOUTH	CLASS NO. 3		10 INVENTORY ENTRIES
	7 WEST	CLASS NO. 4		10 INVENTORY ENTRIES
	9 FLAT	CLASS NO. 5		0 INVENTORY ENTRIES

2.4. PERIPHERAL DATA FILE

2.4. PERIPHERAL DATA FILE

2.4.1. Purpose

The Peripheral Data File is created by PROGRAM SEARCH and contains inventory variable descriptions and values retrieved from the Inventory Data File. The Library user accesses PROGRAM SEARCH to retrieve selected inventory data. The retrieved data meet user-specified constraints and are assigned to user-defined stratification classes.

The Peripheral Data File may then be accessed by PROGRAM SEARCH to compute descriptive statistics (see Section 3.6.), or may be interfaced with other standardized non-Library statistical packages or user application programs. In short, the Peripheral Data File provides the Library user with data manipulation options not available through the six MSP programs.

Section 3.6. explains how to create a Peripheral Data File from the Inventory Data File using MSP PROGRAM SEARCH. One Peripheral Data File is built for every SEARCH command in PROGRAM SEARCH's input stream, and several data files may therefore exist on the same logical device.

2.4.2. Structure

The Peripheral Data File consists of 4 record types:

I: File Size (4 byte integer)

II: Variable Description (64 numeric and alphameric bytes)

III: Inventory Entry Description (32 numeric bytes)

IV: Variable Value (8 byte integer).

If the PROGRAM SEARCH input stream contains several runs (e.g., more than one SEARCH command), several Peripheral Data Files will be written onto the same logical device. In such cases:

- the number of Peripheral Data Files created is equal to the number of PROGRAM SEARCH runs (e.g., the number of SEARCH commands
- the Peripheral Data Files appear in the order in which they were requested in the PROGRAM SEARCH input stream
- each Peripheral Data File is terminated by an end of file mark

2.4.2.1. Peripheral Data File Record Type I - File Size

FILE NAME Peripheral Data File

RECORD NAME File Size

RECORD TYPE I OF IV

RECORD LENGTH 8 BYTES

DESCRIPTION Contains the number of inventory variables retrieved from each inventory entry by PROGRAM SEARCH. If the Library user had requested duff fuel load, dead and down fuel load, and live fuel load (using the TABLE VAR command in PROGRAM SEARCH), then 3 inventory variables will be retrieved from each inventory entry on the Inventory Data File. For this example, the Peripheral Data File size would equal 00000003.

COLUMNS	FORMAT	VARIABLE NAME	DESCRIPTION
1 - 8	I8	NRETRV	Number of retrieved inventory variable values per inventory entry Equals the number of TABLE VAR commands appearing in PROGRAM SEARCH input stream

2.4.2.2. Peripheral Data File Record Type II - Variable Description

FILE NAME Peripheral Data File

RECORD NAME Variable Description

RECORD TYPE II OF IV

RECORD LENGTH 64 BYTES

DESCRIPTION The Variable Description records appear in a contigous block immediately following the File Size record. There are NRETRV (see record type I) Variable Description records, each containing the variable's 4 byte integer code, signed 4 byte base 10 exponent conversion factor, and 56 byte alphameric variable description.

COLUMNS	FORMAT	VARIABLE NAME	DESCRIPTION
1 - 4	I4	KRETRV	4 byte integer variable code
5 - 8	I4	MFACTR	signed 4 byte integer base 10 exponent conversion factor for variable KRETRV
9 - 64	14A4	ARETRV(14)	56 byte alphameric variable description (from the Variable Index File)

2.4.2.3. Peripheral Data File Record Type III - Inventory Entry Description

FILE NAME Peripheral Data File

RECORD NAME Inventory Entry Description

RECORD TYPE III OF IV

RECORD LENGTH 32 BYTES

DESCRIPTION The Inventory Entry Description record appears once for every inventory entry whose variables were retrieved from the Inventory Data File by PROGRAM SEARCH. It is followed by NRETRV (see record type I) Variable Value records (see record type IV) for the entry. The Inventory Entry Description includes entry's key, stand number, number of plots represented by the entry (e.g., the entry's weighting factor), and the entry's position in the stratification scheme defined in PROGRAM SEARCH.

COLUMNS	FORMAT	VARIABLE NAME	DESCRIPTION
1 - 8	I8	KEY	Inventory entry's key
9 - 16	I8	IDSTND	Inventory entry's stand number
17 - 24	I8	NPLOTS	Number of inventory plots represented by the entry (e.g., the entry's weighting factor for weighted statistics)
25 - 28	I4	KCELL(1)	Entry's row stratification class assignment (equals 1 if no row stratification was requested)
29 - 32	I4	KCELL(2)	Entry's column stratification class assignment (equals 1 if no column stratification was requested in PROGRAM SEARCH).

2.4.2.4. Peripheral Data File Record Type IV - Variable Value

FILE NAME Peripheral Data File

RECORD NAME Variable Value

RECORD TYPE IV OF IV

RECORD LENGTH 8 BYTES

DESCRIPTION NRETRV (see record type I) Variable Value records appear after every Inventory Entry Description record (see record type III), and contains the 8 byte integer variable value retrieved from the Inventory Data File.

COLUMNS	FORMAT	VARIABLE NAME	DESCRIPTION
1 - 8	I8	KVALUE	Variable value (for continuous variables) or class (for ordinal variables)

2.4.3. File Maintenance and Use

2.4.3.1. Write Access

The Peripheral Data File is built by PROGRAM SEARCH, and includes 1 data file for every SEARCH command. All Peripheral Data Files created by PROGRAM SEARCH reside on the same logical device.

2.4.3.2. Read Access

PROGRAM SEARCH also read-accesses the Peripheral Data File during each SEARCH to perform statistical computations requested by the Library user.

The Peripheral Data File provides Library users a means to interface NFIL data with external routines. The external routines may be standardized statistical packages or Library user applications programs.

It is important to note that the Peripheral Data Files are temporary files; they will be lost after termination of PROGRAM SEARCH unless the Library user takes the appropriate steps to save the files in the job control stream.

2.3.4. Peripheral Data File Sample Listing

A sample listing of the Peripheral Data File appears in Figure 2.11. It was created by the PROGRAM SEARCH input stream illustrated in Figure 3.24. requesting a single SEARCH for 6 inventory variables. Record types I, II, III, and IV are indicated in the left hand margin.

The first record indicates that 6 inventory variables were recieved. The next 6 records give the variable codes, base 10 exponent conversion factors, and descriptions of the requested variables (record type II). The first inventory entry retrieved has a key of 00000001, stand number 00000001, and represents 1 plot (record type III). It was assigned to row stratification class number 2 and column stratification class number 1. Record type III is then followed by 6 variable values, 1 for each of the six requested variables.

FIGURE 2.11. PERIPHERAL DATA FILE SAMPLE LISTING

```
RECORD TYPE I                6
RECORD TYPE II             411  -30.25 - 1.0 INCH DEAD + DOWN FUEL LOAD
                           421  -31.0 - 3.0 INCH DEAD AND DOWN FUEL LOAD
                           431  -3> 3.0 INCH DEAD AND DOWN FUEL LOAD
                           441  -3TOTAL > 0.25 INCH DEAD AND DOWN FUEL LOAD
                           501  -1HEIGHEST INTERCEPT < 3 INCH DEAD AND DOWN FUEL
                           601  -4DEAD AND DOWN FUEL BULK DENSITY (< 3 INCH PARTICLES)
RECORD TYPE III              1        1        1   2   1
                           449
RECORD TYPE IV            2467
                        133644
                        136560
RECORD TYPE IV              12
                         13369
RECORD TYPE III              2        1        1   2   1
                           536
RECORD TYPE IV            3079
                         95124
                         98739
RECORD TYPE III             14
                         14227
                             3        1        1   2   1
RECORD TYPE IV             489
                           865
                         58071
                         59425
RECORD TYPE III              9
                          8289
                             4        1        1   2   1
RECORD TYPE IV             349
                          3048
                        100635
                        104082
RECORD TYPE III              6
                         31654
                             5        1        1   2   1
RECORD TYPE IV             604
                          2285
                         79962
                         82851
RECORD TYPE III             16
                          9948
                             6        1        1   2   1
RECORD TYPE IV             600
                          1747
                        135329
                        137676
RECORD TYPE III              5
                         25863
                             7        1        1   2   1
RECORD TYPE IV             326
                          1529
                         99873
                        101728
RECORD TYPE III              9
                         11356
                             8        1        1   2   1
RECORD TYPE IV             540
                          2097
                        170558
                        173195
                             10
                         14529
                              9       1        1   2   1
                            401
```

3. MGMT SOFTWARE PCKGE

M

S

P

3. MANAGEMENT SOFTWARE PACKAGE

3.1. PROGRAM INDEX

3.1. PROGRAM INDEX

3.1.1. Purpose

PROGRAM INDEX is an NFIL file maintenance routine. It is used to create a new Variable Index File or to add new variables or variable classes to an existing Variable Index File.

The use of PROGRAM INDEX is the first step in creating a local Fuels Inventory Database. The term 'local' refers to any database created by an off-site or terminal user for his own purposes. The term 'centralized' refers to any database maintained on-site for nation-wide use and application.

Before entering new fuels inventory information into the Inventory Data File, all new variable and variable class codes must first be added to a Variable Index File. The Variable Index File serves both the Library user and MSP programs as a cross reference between variable codes and their descriptions.

A new Variable Index File is built via PROGRAM INDEX using the CREATE facility. The Library user assigns four-digit codes to all variables recorded by the inventory, and to all classes of each variable. PROGRAM INDEX reads the card image input and builds the new Variable Index File.

Whenever new variables or variable classes are introduced to the database, their codes are entered into the existing Variable Index File using the PROGRAM INDEX ADD facility.

3.1.2. Program Input

PROGRAM INDEX operates under two alternative modes:

- CREATE
- ADD

The CREATE mode builds a new Variable Index File containing the variable and variable class codes specified in the input. The ADD mode enters new variable and variable class codes into an existing Variable Index File.

The term 'variable' as used in this documentation refers to any type of inventory factor recorded or measured in the field and to be entered into the database. Four generic variable groups are common to most inventory fuels data:

- location variables (e.g., state, national forest)
- site variables (e.g., elevation, aspect)
- stand variables (e.g., forest cover type, stand age)
- fuel variables (e.g., fuel load, bulk density, percent cover)

Inventory variables may furthermore be divided into three statistical categories:

- continuous variables (e.g., fuel load, elevation)
- count variables (e.g., number of plots represented by an inventory entry)
- ordinal variables (e.g., forest cover type class, national forest)

Each inventory variable has one or more classes assigned to it. In the case of continuous or count variables, a single class must be defined by the Library user. The class code must be 9999, and its description is the units of measure of the variable.

In the case of ordinal variables, one or more classes are defined with codes greater than 0 and less than 9999. For the variable 'national forest', Bitterroot, Clearwater, and Lolo are three potential classes.

3.1.2.1. CREATE-ing a New Variable Index File

A sample PROGRAM INDEX input stream to create a new Variable Index File is illustrated in Figure 3.1. Program input rules (as demonstrated by Figure 3.1) are:

1 The first input card image must contain the command CREATE beginning in column 1 (see line 1).

2 One input card image must appear for every variable and every variable class to be entered into the Variable Index File.

In Figure 3.1., 34 input card images follow the CREATE command. New variables account for 15 of the images, while new variable classes make up the remaining 19.

3 Each input card image includes a variable code, a variable class code, a variable value base 10 exponent conversion factor, and an alphameric description.

The variable on line 2, Figure 3.1. has been assigned a code of 0101, a class code of 0000, a base 10 exponent conversion factor of 0000, and a description 'USDA NATIONAL FOREST'.

4 Input card image format is identical for variables and variable classes.

The input format for 'USDA NATIONAL FOREST' (line 2) is the same as the variable's classes on lines 3 and 4.

5 Columns 1 - 4 must be blank.

6 The variable code is right-justified in columns 5 - 8.

-- variable codes are assigned by the Library user

-- variable codes have a maximum of four digits

-- variable codes 0001 - 0005 are reserved for Library use only

-- an upper limit may be imposed upon legal variable code values (see Section 3.1.4., Program Limitations)

FIGURE 3.1. PROGRAM INDEX 'CREATE' MODE SAMPLE INPUT STREAM

```
LINE          1         2         3         4         5         6         7         8
 NO   12345678901234567890123456789012345678901234567890123456789012345678901234567890
  1   CREATE
  2         101   0   0USDA NATIONAL FOREST
  3         101   3   0BITTERROOT NATIONAL FOREST
  4         101  16   0LOLO NATIONAL FOREST
  5         111   0   0STATE
  6         111  26   0MONTANA
  7         201   0   0FOREST COVER TYPE (BY PLURALITY OF SPECIES)
  8         201 108   0LODGEPOLE PINE
  9         201 202   0DOUGLAS-FIR
 10         211   0   0STAND AGE
 11         2119999   0YEARS
 12         301   0   0ELEVATION
 13         3019999   0FEET ABOVE MEAN SEA LEVEL
 14         411   0  -30.25 - 1.0 INCH DEAD + DOWN FUEL LOAD
 15         4119999  -3TONS/ACRE
 16         421   0  -31.0 - 3.0 INCH DEAD AND DOWN FUEL LOAD
 17         4219999  -3TONS/ACRE
 18         431   0  -3> 3.0 INCH DEAD AND DOWN FUEL LOAD
 19         4319999  -3TONS/ACRE
 20         441   0  -3TOTAL > 0.25 INCH DEAD AND DOWN FUEL LOAD
 21         4419999  -3TONS/ACRE
 22         501   0  -1HEIGHEST INTERCEPT < 3 INCH DEAD AND DOWN FUEL
 23         5019999  -1INCHES
 24         501   0  -4DEAD AND DOWN FUEL BULK DENSITY (< 3 INCH PARTICLES)
 25         5019999  -4LBS/CU FT
 26         701   0   0DUFF FUEL LOAD DETERMINATION METHOD
 27         701   1   0DESTRUCTIVE SAMPLE, OVENDRY AND WEIGH
 28         702   0  -3DUFF FUEL LOAD
 29         7029999  -3TONS/ACRE
 30         801   0  -3TOTAL FUEL LOAD
 31         8019999  -3TONS/ACRE
 32         221   0   0STAND MGMT STATUS (FUEL TREATMENT)
 33         221   1   0OLD GROWTH STAND
 34         221   2   0SECOND GROWTH STOCKED STAND
 35         221  11   0CLEARCUT LOGGING SLASH
```

7 Variable class codes are right-justified in columns 9 - 12.

-- variable input images must have a class code set equal to 0 or blank

The variable 'USDA NATIONAL FOREST' on line 2 has a class code of 0.

-- one class only must be assigned to a continuous or count variable, and its value must be 9999

The variable 'STAND AGE' on line 10 has a single class on line 11. The class code for the variable is set equal to 0, and the class code for the variable class is set equal to 9999.

-- variable class codes must be greater than 0 and less than 9999 for ordinal variables

'USDA NATIONAL FOREST' is an ordinal variable with two classes. 'BITTERROOT NATIONAL FOREST' has a class code of 3, and 'LOLO NATIONAL FOREST' has a code of 16. The codes need not be sequential, and may follow some scheme useful to the Library user. In this case, Region One national forest codes are used. For the variable 'FOREST COVER TYPE' on lines 7 - 9, SAF species codes are used as class codes.

8 The variable value base 10 exponent conversion factor is right-justified in columns 13 - 16.

-- must equal 0 for ordinal and count variable and variable classes

The base 10 conversion factor for 'USDA NATIONAL FOREST' and its classes is set equal to 0, as for the variable 'FOREST COVER TYPE' and its classes on lines 7 - 9.

-- must be less than 0 for continuous variables

The variable 'DUFF FUEL LOAD' on line 28 has a conversion factor of -3. This will convert the duff load value 12345678 as stored in the Inventory Data File to 12345.678 tons per acre, as defined on line 29.

-- the conversion factor must be identical for all classes of a variable, and must also appear on the variable input image

The 'DUFF FUEL LOAD' conversion factor -3 is the same on line 28 (the variable input card) and line 29 (the variable class input card).

9 Variable and variable class descriptions appear left-justified in columns 17 - 80.

-- the most significant portion of the alphameric description should appear in the first 40 columns (columns 17 - 56)

-- the description should include the units of measure for continuous variable classes

The variable 'STAND AGE' on line 10 has a single class on line 11. The class description documents the units of measure, 'YEARS'. Similarly, for 'DUFF FUEL LOAD' class 9999 on line 29, the description is 'TONS PER ACRE'.

10 Every variable must have at least one class assigned to it.

11 The variable input card image must appear before any of the variable's class input card images.

The following recommendations are made to NFIL users in addition to the above rules:

1 The card input stream should be ordered first by variable code, and second by variable class code. Pre-ordering will enhance program execution.

Lines 32 - 35 of Figure 3.1. illustrates a variable and its classes that were not pre-ordered in the input stream. While this will not cause a fatal error, it will increase program execution time by a slight amount.

2 While it is not required that variable classes follow immediately behind their associated variable input card, doing so will optimize program execution and reduce the chances of violating input rule number 11.

3 Be sure to include a variable and classes describing the fuels treatment on the inventory site. Lines 32 - 35 demonstrate a sample variable 'STAND MGMT STATUS (FUEL TREATMENT)' that will enable the Library user to identify fuel condition and past treatment.

4 Include variables for subtotals, such as those on lines 20 and 30. If not included, PROGRAM SEARCH must be interfaced with an external applications program to get totals and subtotals.

5 Include variables indicating the method of assessment for companion variables. For example, the variable 'DUFF FUEL LOAD DETERMINATION METHOD' on line 26 will identify the method used to determine 'DUFF FUEL LOAD' on line 28. Such information may be of use later on in controlling the quality of data retrieved by PROGRAM SEARCH. By way of example, the Library user may wish to retrieve only those duff fuel loads determined by destructive sampling methods rather than those based on linear regressions on duff depth.

3.1.2.2. ADD-ing to an Existing Variable Index File

A sample PROGRAM INDEX input stream to add new variables and variable classes to an existing Variable Index File is illustrated in Figure 3.2. The sample input stream represents an addition of new variables and variable classes to the Variable Index File created in Figure 3.1.

The rules for ADDing information are the same as those for CREATEing a file, except:

1 The first input card image must contain the command ADD beginning in column 1 (see line 1, Figure 3.2.).

2 New variable class input card images may be included for variables already on the Variable Index File.

The new variable class 'CLEARWATER NATIONAL FOREST' with class code 0005 (line 2) will be added to the 'USDA NATIONAL FOREST' variable class list. Similarly, 'WESTERN RED CEDAR' (line 4) will be added to the 'FOREST COVER TYPE' class list, and 'REGRESSION ON DUFF DEPTH MEASUREMENT' (line 11) will be included in the 'DUFF FUEL LOAD DETERMINATION METHOD' class list.

3 Any new variables and their classes may be assigned.

The new variable 'ASPECT' is declared on line 5 with a code of 0311 and 5 associated classes. New variables are also declared on lines 12, 14, and 16.

4 Variable codes and variable class codes may not be reassigned under any circumstances. To do so would create a disparity between Variable Index Codes and the variable codes already stored on the Inventory Data File.

FIGURE 3.2. PROGRAM INDEX 'ADD' MODE SAMPLE INPUT STREAM

```
LINE          1         2         3         4         5         6         7         8
 NO   12345678901234567890123456789012345678901234567890123456789012345678901234567890
  1   ADD
  2         101   5   0CLEARWATER NATIONAL FOREST
  3         111  12   0IDAHO
  4         201 242   0WESTERN RED CEDAR
  5         311   0   0ASPECT
  6         311   1   0NORTH
  7         311   3   0EAST
  8         311   5   0SOUTH
  9         311   7   0WEST
 10         311   9   0FLAT
 11         701  11   0REGRESSION ON DUFF DEPTH MEASUREMENT
 12         711   0  -1DUFF DEPTH
 13         7119999  -1INCHES
 14         721   0  -4DUFF FUEL BULK DENSITY
 15         7219999  -4LBS/CU FT
 16         901   0  -2SHRUB COVER
 17         9019999  -2FRACTION OF GROUND AREA COVERED
```

3.1.3. Program Output

PROGRAM INDEX output generated by the sample input stream in Figure 3.1. is shown in Figures 3.3. through 3.5.

A title page is output first, showing the date and time PROGRAM INDEX was submitted for execution (Figure 3.3.).

The title page is followed by one or more pages echoing the input card images and indicating the action taken by PROGRAM INDEX for each image (Figure 3.4.). As each input card image is read, it is echoed back to the lineprinter. The image is then checked for correct syntax. If the input image is in correct form and does not violate any of the input rules, a message is printed indicating whether a new variable was added to the Variable Index File, or whether a new class was added to an existing variable. If an input error was detected, an error message is generated (see Section 3.1.5. and Figure 3.7.).

PROGRAM INDEX output is concluded with a closing page indicating whether or not the input stream was accepted and the number of codes entered into the Variable Index File (Figure 3.5.).

Output generated by the input stream illustrated in Figure 3.2. is shown in Figure 3.6.

The Variable Index File resulting from the input streams in Figures 3.1. and 3.2. is illustrated in Figure 2.9.

FIGURE 3.3. PROGRAM INDEX SAMPLE TITLE PAGE OUTPUT

```
NATIONAL FUELS INVENTORY LIBRARY

**** PROGRAM INDEX ****
VERSION 01
LAST REVISED ON JUN 22, 1979

DEVELOPED BY
SYSTEMS FOR ENVIRONMENTAL MGMT
P.O. BOX 3775
MISSOULA, MONTANA 59806

FOR THE
NATIONAL FUELS INVENTORY AND APPRAISAL PROJECT
ROCKY MOUNTAIN FOREST AND RANGE EXPERIMENT STATION
FORT COLLINS, COLORADO

BEGIN ON
DATE :  22 SEP 79
TIME :  12.15.12
```

FIGURE 3.4. PROGRAM INDEX 'CREATE' MODE SAMPLE OUTPUT

```
CREATE

***** NEW VARIABLE INDEX FILE LABEL, FILE SIZE, AND VARIABLES 1- 3 HAVE BEEN CREATED.

     101   0   USDA NATIONAL FOREST
NEW VARIABLE CODE  101 WILL BE ADDED TO THE INDEX FILE

     101   3   USBITTERROOT NATIONAL FOREST
NEW CLASS    3 WILL BE ADDED TO EXISTING VARIABLE CODE  101

     101  15   UCOLO NATIONAL FOREST
NEW CLASS   15 WILL BE ADDED TO EXISTING VARIABLE CODE  101

     111   0   USTATE
NEW VARIABLE CODE  111 WILL BE ADDED TO THE INDEX FILE

     111  25   UMONTANA
NEW CLASS   25 WILL BE ADDED TO EXISTING VARIABLE CODE  111

     201   0   UFOREST COVER TYPE (BY PLURALITY OF SPECIES)
NEW VARIABLE CODE  201 WILL BE ADDED TO THE INDEX FILE

     201 108   ULODGEPOLE PINE
NEW CLASS  108 WILL BE ADDED TO EXISTING VARIABLE CODE  201

     201 212   UDOUGLAS-FIR
NEW CLASS  212 WILL BE ADDED TO EXISTING VARIABLE CODE  201

     211   0   USTAND AGE
NEW VARIABLE CODE  211 WILL BE ADDED TO THE INDEX FILE

     211 9999   UYEARS
NEW CLASS 9999 WILL BE ADDED TO EXISTING VARIABLE CODE  211

     301   0   UELEVATION
NEW VARIABLE CODE  301 WILL BE ADDED TO THE INDEX FILE

     301 9999   UFEET ABOVE MEAN SEA LEVEL
NEW CLASS 9999 WILL BE ADDED TO EXISTING VARIABLE CODE  301

     +11   0   -30.25 - 1.0 INCH DEAD + DOWN FUEL LOAD
NEW VARIABLE CODE  411 WILL BE ADDED TO THE INDEX FILE

     +11 9999   -31 TONS/ACRE
NEW CLASS 9999 WILL BE ADDED TO EXISTING VARIABLE CODE  411

     +21   0   -31.0 - 3.0 INCH DEAD AND DOWN FUEL LOAD
NEW VARIABLE CODE  421 WILL BE ADDED TO THE INDEX FILE

     +21 9999   -31 TONS/ACRE
NEW CLASS 9999 WILL BE ADDED TO EXISTING VARIABLE CODE  421

     +31   0   -32 3.0 INCH DEAD AND DOWN FUEL LOAD
NEW VARIABLE CODE  431 WILL BE ADDED TO THE INDEX FILE

     +31 9999   -31 TONS/ACRE
NEW CLASS 9999 WILL BE ADDED TO EXISTING VARIABLE CODE  431

     +41   0   -31 TOTAL > 0.25 INCH DEAD AND DOWN FUEL LOAD
NEW VARIABLE CODE  441 WILL BE ADDED TO THE INDEX FILE
```

FIGURE 3.5. PROGRAM INDEX SAMPLE CLOSING PAGE OUTPUT

**** PROGRAM INDEX ****

TERMINATED SUCCESSFULLY WITH 0 FATAL ERRORS

14 VEG CODES WERE READ IN AND WRITTEN TO THE VARIABLE CODE INDEX FILE

CLOSE ON
22 SEP 79
12.16.16

FIGURE 3.6. PROGRAM INDEX 'ADD' MODE SAMPLE OUTPUT

```
ADD
     101   3   0CLEARWATER NATIONAL FOREST
NEW CLASS     3 WILL BE ADDED TO EXISTING VARIABLE CODE  101

     111  12   01DAHO
NEW CLASS    12 WILL BE ADDED TO EXISTING VARIABLE CODE  111

     201 242   0WESTERN RED CEDAR
NEW CLASS   242 WILL BE ADDED TO EXISTING VARIABLE CODE  201

     311   0   0ASPECT
NEW VARIABLE CODE  311 WILL BE ADDED TO THE INDEX FILE

     311   1   0NORTH
NEW CLASS     1 WILL BE ADDED TO EXISTING VARIABLE CODE  311

     311   3   0EAST
NEW CLASS     3 WILL BE ADDED TO EXISTING VARIABLE CODE  311

     311   5   0SOUTH
NEW CLASS     5 WILL BE ADDED TO EXISTING VARIABLE CODE  311

     311   7   0WEST
NEW CLASS     7 WILL BE ADDED TO EXISTING VARIABLE CODE  311

     311   9   0FLAT
NEW CLASS     9 WILL BE ADDED TO EXISTING VARIABLE CODE  311

     701  11   0REGRESSION ON DUFF DEPTH MEASUREMENT
NEW CLASS    11 WILL BE ADDED TO EXISTING VARIABLE CODE  701

     711   0   -1DUFF DEPTH
NEW VARIABLE CODE  711 WILL BE ADDED TO THE INDEX FILE

     711****   -1INCHES
NEW CLASS **** WILL BE ADDED TO EXISTING VARIABLE CODE  711

     721   0   -4DUFF FUEL BULK DENSITY
NEW VARIABLE CODE  721 WILL BE ADDED TO THE INDEX FILE

     721****   -4LBS/CU FT
NEW CLASS **** WILL BE ADDED TO EXISTING VARIABLE CODE  721

     901   0   -2SHRUB COVER
NEW VARIABLE CODE  901 WILL BE ADDED TO THE INDEX FILE

     901****   -2FRACTION OF GROUND AREA COVERED
NEW CLASS **** WILL BE ADDED TO EXISTING VARIABLE CODE  901
```

3.1.4. Program Limitations

3.1.4.1. Lawrence Berkeley Laboratory CDC 6000/7000

The following limitations are in effect for the LBL CDC 6000/7000 version of PROGRAM INDEX:

1 Variable codes may not exceed an assigned value of 1500

2 A maximum of 500 variables may be assigned to the Variable Index File

3.1.4.2. Fort Collins Computer Center UNIVAC 1100/82

The following limitations are in effect for the FCCC UNIVAC version of PROGRAM INDEX:

1 Variable codes may not exceed an assigned value of 1500

2 A maximum of 500 variables may be assigned to the Variable Index File

3.1.5. Program Error Messages

PROGRAM INDEX will detect five types of input errors and print diagnostics back to the lineprinter.

Fatal Error No. 1 occurs whenever the first input card image is not an ADD or CREATE command (violation of Rule 1).

Fatal Error No. 2 occurs whenever the user assigns a variable code which exceeds the allowable maximum value.

Fatal Error No. 3 occurs whenever a variable class input image appears ahead of its associated variable input image (Rules 7 and 11).

Fatal Error No. 4 occurs whenever the user assigns a variable code less than or equal to 0005. Variables codes 0001 through 0005 are reserved for Library use only (Rule 6).

Fatal Error No. 5 occurs when PROGRAM INDEX reviews the existing Variable Index File and the new input stream and detects a variable with no classes assigned to it (Rule 7).

Any fatal errors detected by PROGRAM INDEX in the input stream will affect the following actions:

- the remaining input card images will be checked for correct form
- none of the new variables and variable classes in the input stream will be added to the Variable Index File
- the existing Variable Index File will be restored

Figure 3.7. illustrates a PROGRAM INDEX input stream which has caused all 5 diagnostic messages to appear. The closing output page for the run is shown in Figure 3.8. The first error was caused by the omission of the CREATE/ADD command. Since PROGRAM INDEX attempted to interpret variable code 1001 as the command, the variable's first class record was rejected.

```
          1001    )  -4TOTAL LIVE FUEL LOAD

FATAL ERROR NO. 1  --  UNRECOGNIZABLE COMMAND.  FIRST CARD MUST BE +ADD+ OR +CREATE+ STARTING IN COL 1.

          10019999  -4TONS/ACRE

FATAL ERROR NO. 3  --  NEW VARIABLES MUST HAVE A CLASS CODE OF 0. AND NEW VARIABLES MUST APPEAR BEFORE
                       THEIR CLASSES IN THE INPUT STREAM.

          1200   0  -4TOTAL DEAD + LIVE FUEL LOAD
     NEW VARIABLE CODE 1200 WILL BE ADDED TO THE INDEX FILE

          1500   0  -1SHRUB HEIGHT

FATAL ERROR NO. 2  --  ASSIGNED VARIABLE CODE EXCEEDS MAXIMUM PERMISSABLE VALUE OF 1500

          15009999  -1FEET

FATAL ERROR NO. 2  --  ASSIGNED VARIABLE CODE EXCEEDS MAXIMUM PERMISSABLE VALUE OF 1500

FATAL ERROR NO. 5  --  VARIABLE CODE 1200 HAS NO CLASSES ASSIGNED TO IT.

             2   0   0USDA FOREST SERVICE REGION

FATAL ERROR NO. 4  --  ASSIGNED VARIABLE CODE IS NOT GREATER THAN 0004.
```

FIGURE 3.7. PROGRAM INDEX ERROR MESSAGES

FIGURE 3.8. PROGRAM INDEX SAMPLE CLOSING PAGE OUTPUT FOR INPUT STREAM WITH SYNTAX ERRORS

```
**** PROGRAM INDEX ****

TERMINATED UNSUCCESSFULLY  WITH   5 FATAL ERRORS

1 NEW CODES WERE READ IN AND NOT WRITTEN TO THE VARIABLE CODE INDEX FILE

CLOSE ON
29 SEP 79
13.12.47
```

3.1.6. Program Structure and Logic

PROGRAM INDEX consists of a MAIN routine, 7 subroutines, and a BLOCK DATA routine.

3.1.6.1. MAIN

- Step 1: Call SUBROUTINE HEADER to write the output title page.
- Step 2: Read first input card image to determine whether a CREATE or ADD is requested.
- Step 3: If the first input card image is not CREATE or ADD, call SUBROUTINE MESAGE to print error diagnostics.
- Step 4: If CREATE, call SUBROUTINE CREATE to write a new Variable Index File label (record type I), file size (record type II), and first three variables and variable classes (record types III).
- Step 5: Call SUBROUTINE OLDIND to read and store variable types and codes on the existing Variable Index File. In the case of a CREATE run, the existing Variable Index File is that created in Step 4.
- Step 6: Call SUBROUTINE READIN to read and check the remaining input stream. If input images are accepted, they are written to logical unit N2.
- Step 7: If no fatal errors are detected, SUBROUTINE NEWIND is called to merge the new variables and variable classes with the existing variables and variable classes.
- Step 8: SUBROUTINE CLOSER is called to write the sign-off output page.
- Step 9: END.

3.1.6.2. SUBROUTINE HEADER

- Step 1: Write PROGRAM INDEX heading on output title page.
- Step 2: Write date and time PROGRAM INDEX was submitted on output title page.
- Step 3: RETURN to MAIN.

3.1.6.3. SUBROUTINE CREATE

- Step 1: Write a Variable Index File label (record type I) and the date to logical device N1.
- Step 2: Write the Variable Index file size (record type II) to logical device N1. The file size is equal to 00000003.
- Step 3: Write variables 0001 (NFIL inventory entry key), 0002 (NFIL stand number), and 0003 (number of plots represented by the inventory entry) and their single classes to logical device N1 (record type III).
- Step 4: REWIND N1.
- Step 5: Write output message indicating that a new Variable Index File will be created.
- Step 6: RETURN to MAIN.

3.1.6.4. SUBROUTINE OLDIND

- Step 1: Read the existing Variable Index File label (record type I) and file size (record type II) from logical device N1.
- Step 2: Read each variable code (record type III) and its number of classes from logical device N1.
- Step 3: Convert each variable's user-assigned four-digit code to a number indicating the variable's order in the file.
- Step 4: REWIND N1.
- Step 5: RETURN to MAIN.

3.1.6.5. SUBROUTINE READIN

- Step 1: Read each input card image's variable code (KODE), variable class (KLASS), and base 10 exponent conversion factor (MFACT). If end of input, go to Step 9.
- Step 2: If KODE is greater than the allowable maximum (MXKODE), call SUBROUTINE MESAGE to print an error diagnostic.
- Step 3: If KODE is less than 0006, call SUBROUTINE MESAGE to print an error doagnostic.
- Step 4: If the input image is for a new class to an existing variable, increment the variable's class counter NCLASS and call SUBROUTINE MESAGE to print PROGRAM INDEX's interpretation of the input image.
- Step 5: If the input variable code was not in the existing Variable Index File, a new variable type is assumed. If its class code is not equal to 0, SUBROUTINE MESGAE is called to print an error diagnostic. If accepted, the new variable type is assigned a sequence number and the Variable Index file size (record type II) is incremented.
- Step 6: The new input card image's variable code (KODE), class code (KLASS), number of classes or class sequence number (NCLASS(KODE)), base 10 exponent conversion factor (MFACT), and alphameric description (ADOC(I), I=5,18) are written to logical device N2.
- Step 7: Number of new index entries (NINDEX) is incremented.
- Step 8: Next input card image (go to Step 1).
- Step 9: ENDFILE N2.
- Step 10: Check each existing and new variable to ensure each has at least 1 class assigned to it. If not, an error diagnostic is written to output, and the fatal error counter NFATAL is incremented.
- Step 11: RETURN to MAIN.

3.1.6.6. SUBROUTINE MESAGE(I,FATAL)

- Step 1: If FATAL is equal to .TRUE., the fatal error counter NFATAL is incremented.
- Step 2: Control is transferred to statement number I.
- Step 3: Statement I writes a message or error diagnostic as per format statement number 600I.
- Step 4: RETURN to calling routine.

3.1.6.7. SUBROUTINE NEWIND

- Step 1: Read existing Variable Index File label (record type I) and file size (record type II) from logical device N1.
- Step 2: Read existing Variable Index File variable and variable class records (record type III) from logical device N1 and write them to logical device N3.
- Step 3: Read new variables and variable classes (record type III) from logical device N2 and write then to logical device N3.
- Step 4: Rewind logical device N3 and sort all record type III's by variable code, then by variable class code. Sorted records are written onto logical device N4.
- Step 5: Logical device N1 is rewound. The Variable Index File label (record type I) and new file size (record type II) are written onto logical device N1.
- Step 6: Sorted variable and variable class record type III's are read from logical device N4 and written onto logical device N1.
- Step 7: An end of file mark is written onto logical device N1.
- Step 8: RETURN to MAIN.

3.1.6.8. SUBROUTINE CLOSER

- Step 1: Write output message indicating whether PROGRAM INDEX terminated successfully or unsuccessfully. If not successful, the number of fatal errors detected is printed.

- Step 2: The number of new variable and variable codes read from the input stream is written to output along with an indication of whether they were or were not written onto the Variable Index File.

- Step 3: The date and time PROGRAM INDEX terminated is printed.

- Step 4: RETURN to MAIN.

3.1.7. Logical Device Assignment

PROGRAM INDEX uses six logical devices assigned to variables N1, N2, N3, N4, N5, and N6. Logical device numbers are assigned to the variables in the BLOCK DATA subroutine.

DEVICE	USE
N1	Read-access to existing Variable Index File for ADD mode. Write-access to create new file label and size records for CREATE mode.
N2	Write-access for new variable and variable class entries.
N3	Write-access to new, sorted Variable Index File if no fatal errors are detected. Write-access to existing, sorted Variable Index File if fatal errors are detected.
N4	SORT directives (CDC only)
N5	Card image input
N6	Lineprinter output
N7	SORT messages (CDC only)

3.1.7.1. Lawrence Berkeley Laboratory CDC 6000/7000

Logical devices are assigned as follows:

```
DATA N1,N2,N3,N4,N5,N6,N7/1,2,3,4,5,6,7/
```

3.1.7.2. Fort Collins Computer Center UNIVAC 1100/82

Logical devices are assigned as follows:

```
DATA N1,N2,N3,N4,N5,N6,N7/8,2,3,4,5,6,7/
```

3.2. PROGRAM DATA

3.2. PROGRAM DATA

3.2.1. Purpose

PROGRAM DATA is an NFIL file maintenance routine. It is used to create a new Stand Documentation File and Inventory Data File, or to add new inventory information to existing Stand Documentation and Inventory Data files.

Before PROGRAM DATA may be used, all variable and variable class codes not already in the Variable Index File must be entered into the file via PROGRAM INDEX.

PROGRAM DATA reads stand and inventory entry documentation and enters it into the Stand Documentation File. This file contains each stand's Library-assigned stand number, user-supplied documentation, number of inventory entries for the stand, and a list of each inventory entry (see Section 2.2.). Each entry is then assigned a Library key number, and the entry's code:value pairs are added to the Inventory Data File.

The term 'inventory entry' as used in this documentation is defined by a set of code:value pairs. The 'code' identifies the inventory variable whose measure is in the 'value' field. All inventory variable codes are documented in the Variable Index File. An inventory entry may represent a single plot for which code:value pairs were recorded, or it may represent many inventory plots for which the code:value pairs record plot averages. Regardless, an inventory entry is an autonomous set of code:value pairs that are to be entered into the Fuels Inventory Database files.

The term 'stand' as used in this documentation refers to a collection of inventory entries that are assigned a common stand number by PROGRAM DATA. The Library user determines which inventory entries will be assigned to common stands through use of the NEW STAND command. In many cases, a stand may consist of several inventory plots taken in a specific forest stand, but it may also consist of a single inventory entry. The latter occurs most commonly when an inventory entry records stand summary rather than inventory plot data.

3.2.2. Program Input

PROGRAM DATA operates under two alternative modes:

- CREATE
- ADD

The CREATE mode builds a new Stand Documentation File and Inventory Data File containing the inventory information supplied in the input stream. The ADD mode enters new stand documentation and inventory data into existing files.

3.2.2.1. CREATE-ing a New Stand Documentation and Inventory Data File

The first 50 input card images of a sample PROGRAM DATA input stream is illustrated in Figure 3.9. PROGRAM DATA input rules, as demonstrated by the figure, are as follows:

1: The first card input image must contain the command CREATE beginning in column 1 (line 1).

2: A NEW STAND command beginning in column 1 indicates to PROGRAM DATA that all trailing inventory entries are to be assigned to a common stand number. At least 1 inventory entry must follow every NEW STAND command.

The NEW STAND command on line 2, Figure 2.9. indicates that inventory entries ALPHA 1 (line 9), ALPHA 2 (line 25), and ALPHA 3 (line 41) are to be included in a common stand and assigned the same stand number.

3: Each NEW STAND card is followed by any number of stand documentation input card images. Stand documentation input must begin in column 5. All stand documentation cards will appear verbatim in the Stand Documentation File.

The stand documentation shown in lines 3 through 8 is the minimum type and amount recommended to Library users. Information on stand location, stand factors, site factors, fuel type, and inventory date may be supplimented by any additional notes the Library user may find useful.

4: Stand documentation cards are followed by one or more inventory entry data sets. Each inventory entry data set consists of an inventory entry documentation image followed by one input image for every code:value pair. Inventory entry data images must begin in column 9.

5: The inventory entry documentation is the first input image of each inventory entry data set. It includes:

-- the number of code:value pairs for the entry, right-justified in columns 9 - 12

In line 9, inventory entry ALPHA 1 has 15 trailing code:value pairs (lines 10 - 24).

-- the number of inventory plots represented by the inventory entry, right-justified in columns 13 - 16

In line 9, inventory entry ALPHA 1 represents a single inventory plot (e.g., ALPHA 1 does not record summary data).

FIGURE 3.9. PROGRAM DATA 'CREATE' MODE SAMPLE INPUT STREAM

```
LINE          1         2         3         4         5         6         7         8
 NO   12345678901234567890123456789012345678901234567890123456789012345678901234567890
  1   CREATE
  2   NEW STAND
  3       STAND ID - ALPHA
  4       LOCATION - BITTERROOT NF, SULA RD, SW MONTANA, USA
  5       STAND    - SECOND GROWTH DOUGLAS-FIR AFTER 1940 CLEARCUT
  6       SITE     - 3700 FT, NE ASPECT, HIGH PRODUCTIVITY
  7       FUELS    - NATURAL
  8       DATE     - JULY 1977
  9            15   ALPHA  1
 10           101       3
 11           111      26
 12           201     202
 13           211      25
 14           221       2
 15           301    3780
 16           411     449
 17           421    2467
 18           431  133644
 19           441  136560
 20           501      12
 21           601   13389
 22           701       1
 23           702   11119
 24           801  147579
 25            15   ALPHA  2
 26           101       3
 27           111      26
 28           201     202
 29           211      25
 30           221       2
 31           301    3984
 32           411     536
 33           421    3079
 34           431   95124
 35           441   98739
 36           501      14
 37           601   14227
 38           701       1
 39           702   18614
 40           801  117353
 41            15   ALPHA  3
 42           101       3
 43           111      26
 44           201     202
 45           211      25
 46           221       2
 47           301    3437
 48           411     489
 49           421     865
 50           431   58071
```

-- an inventory entry 8 byte alphameric code in columns 17 through 24

In line 9, the inventory entry was called 'ALPHA 1' on the field data sheet.

6: Each inventory entry documentation image is followed by several code:value pair images. The number of trailing code:value images was indicated in columns 9 - 12 of the inventory entry documentation.

-- the variable code appears right-justified in columns 9 - 12

In line 10, the variable code is 0101.

-- the variable value (if continuous or count) or variable class code (if ordinal) appears right-justified in columns 13 - 20

In line 10, variable 0101 ('USDA NATIONAL FOREST', see Figure 3.1.) is ordinal and has a class assignment of 00000003 ('BITTERROOT NATIONAL FOREST').

In line 23, variable 0702 ('DUFF FUEL LOAD') has a value of 00011119, or 11.119 tons per acre.

-- all variable codes used must have previously been entered into the Variable Index File via PROGRAM INDEX

-- all variable values must be 8 byte integers determined as:

K = V / (10**M), where

K is the 8 byte integer value required by PROGRAM DATA, V is the variable's value or class assignment, and M is its base 10 exponent conversion factor

In line 23, variable 0702 has M = -3 (see Figure 3.1., line 28). The measured duff load was 11.1194 tons per acre, and therefore

K = 11.1194 / (10**-3) = 11119.

M equals 0 for all count and ordinal variables, and since 10**0 is unity, variable counts and class assignments may be input directly without using the conversion formula.

-- code:value pairs for each inventory entry should be in ascending order by variable code

3.2.2.2. ADD-ing to Existing Stand Documentation and Inventory Data Files

The rules for ADDing new inventory information to existing Stand Documentation and Inventory Data files are the same as those for the CREATE mode, except:

1: The first card input image must contain the command ADD beginning in column 1.

3.2.2.3. Input Recommendations

The following procedure is recommended for preparing and submitting new inventory data to PROGRAM DATA.

- Step 1: Make a list of all variable types recorded in the inventory.
  - -- include inventory location variables
  - -- include stand, site, and fuel variables
  - -- create new variables, where appropriate, for inventory totals and subtotals
  - -- create new variables, where appropriate, indicating the assessment methods used in the inventory procedure (e.g., create a variable for the method of determining dead and down fuel load)

- Step 2: Enter all new variable and variable class codes into the Variable Index File using PROGRAM INDEX.

- Step 3: Write an applications computer program to convert the raw inventory data into variable code:value pairs.
  - -- program output should be designed as PROGRAM DATA input
  - -- be sure all continuous variable values have been converted to 8 byte right-justified integers as dictated by the variables' base 10 exponent conversion factors

- Step 4: Run the applications program and write output to a storage device.

- Step 5: Run PROGRAM DATA using the storage file as the input device.

3.2.3. Program Output

PROGRAM DATA output generated by the sample input stream in Figure 3.9 is shown in Figures 3.10. through 3.12.

PROGRAM DATA first prints a title page showing the date and time the program was submitted for execution (Figure 3.10.).

The title page is followed by one or more pages echoing the NEW STAND and stand documentation cards for every user-identified stand in the input stream (Figure 3.11.). A line indicating each new stand number, number of inventory entries, and inventory entry keys is also printed after the stand documentation cards.

PROGRAM DATA output is concluded with a closing page indicating whether or not the program terminated successfully and the number of fatal errors detected (Figure 3.12.). The number of new stands and their stand numbers is indicated, as are the number of new inventory entries and their keys.

The Stand Documentation File resulting from the sample input stream of Figure 3.9. is shown in Figure 2.5. The Inventory Data File resulting from the sample input stream is shown in Figure 2.2.

FIGURE 3.10. PROGRAM DATA SAMPLE TITLE PAGE OUTPUT

```
NATIONAL FUELS INVENTORY LIBRARY

**** PROGRAM  DATA ****
VERSION 01
LAST REVISED ON JUN 22, 1979


DEVELOPED BY
SYSTEMS FOR ENVIRONMENTAL MGMT
P.O. BOX 3776
MISSOULA, MONTANA 59806

FOR THE
NATIONAL FUELS INVENTORY AND APPRAISAL PROJECT
ROCKY MOUNTAIN FOREST AND RANGE EXPERIMENT STATION
FORT COLLINS, COLORADO


BEGIN ON
DATE :  22 SEP 79
TIME :  15.15.34
```

FIGURE 3.11. PROGRAM DATA 'CREATE' MODE SAMPLE OUTPUT

```
CREATE

***** NEW STAND DOCUMENTATION AND INVENTORY DATA FILES WERE CREATED.

NEW STAND
    STAND ID - ALPHA
    LOCATION - BITTERROOT NF, SULA RD, SW MONTANA, USA
    STAND    - SECOND GROWTH DOUGLAS-FIR AFTER 1940 CLEARCUT
    SITE     - 3700 FT, NE ASPECT, HIGH PRODUCTIVITY
    FUELS    - NATURAL
    DATE     - JULY 1977
***** ASSIGNED TO NFIL LIBRARY STAND NO.      1 WITH   10 INVENTORY ENTRIES (KEY NOS.    1 -    10)

NEW STAND
    STAND ID - BETA
    LOCATION - BITTERROOT NF, WEST FORK RD, SW MONTANA, USA
    STAND    - OLD GROWTH DOUGLAS-FIR REPLACING DECADENT LODGEPOLE STAND
    SITE     - 5300 FT, EAST ASPECT
    FUELS    - NATURAL
    DATE     - AUGUST 1977
***** ASSIGNED TO NFIL LIBRARY STAND NO.      2 WITH   10 INVENTORY ENTRIES (KEY NOS.   11 -    20)

NEW STAND
    STAND ID - GAMMA
    LOCATION - BITTERROOT NF, DARBY RD, SW MONTANA, USA
    STAND    - EVEN-AGED LODGEPOLE PINE, 75 YEARS AFTER FIRE
    SITE     - 7400 FT, SOUTH ASPECT
    FUELS    - NATURAL
    DATE     - AUGUST 1977
***** ASSIGNED TO NFIL LIBRARY STAND NO.      3 WITH   10 INVENTORY ENTRIES (KEY NOS.   21 -    30)

NEW STAND
    STAND ID - ALPHA
    LOCATION - LOLO NF, SEELEY LAKE RD, WEST-CENTRAL MONTANA, USA
    STAND    - OLD GROWTH DOUGLAS-FIR REPLACING DECADENT LODGEPOLE PINE
    SITE     - 4300 FT, NW ASPECT
    FUELS    - NATURAL
    DATE     - JULY 1978
***** ASSIGNED TO NFIL LIBRARY STAND NO.      4 WITH   10 INVENTORY ENTRIES (KEY NOS.   31 -    40)

NEW STAND
    STAND ID - BETA
    LOCATION - LOLO NF, MISSOULA RD, WEST-CENTRAL MONTANA, USA
    STAND    - LODGEPOLE PINE 80 YEARS AFTER WILDFIRE
    SITE     - 5500 FT, SW ASPECT
    FUELS    - NATURAL
    DATE     - SEPTEMBER 1978
***** ASSIGNED TO NFIL LIBRARY STAND NO.      5 WITH   10 INVENTORY ENTRIES (KEY NOS.   41 -    50)

NEW STAND
    STAND ID - GAMMA
    LOCATION - LOLO NF, SUPERIOR RD, WEST-CENTRAL MONTANA, USA
    STAND    - LODGEPOLE PINE 125 AFTER WILDFIRE
    SITE     - 5500 FT, EAST ASPECT
    FUELS    - NATURAL
    DATE     - AUGUST 1978
***** ASSIGNED TO NFIL LIBRARY STAND NO.      6 WITH   10 INVENTORY ENTRIES (KEY NOS.   51 -    60)
```

FIGURE 3.12. PROGRAM DATA SAMPLE CLOSING PAGE OUTPUT

```
**** PROGRAM  DATA ****

TERMINATED SUCCESSFULLY  WITH  0 FATAL ERRORS

 5 NEW STANDS (NOS.      1 -      6) WERE READ AND STORED .

60 NEW PLOTS  (NOS.      1 -     60) WERE READ AND STORED .


CLOSE ON
22 SEP 79
15.15.36
```

3.2.4. Program Limitations

3.2.4.1. Lawrence Berkeley Laboratory CDC 6000/7000

The following limitations are in effect for the LBL CDC 6000/7000 version of PROGRAM INDEX:

1: Variable codes may not exceed an assigned value of 1500

2: Variable codes may not be less than an assigned value of 6

3: The number of NEW STAND cards in a single input stream may not exceed 10,000

4: The number of stand documentation cards per stand may not exceed 1,000

5: The number of inventory entries per stand may not exceed 10,000

3.2.4.2. Fort Collins Computer Center UNIVAC 1100/82

The following limitations are in effect for the FCCC UNIVAC 1100 version of PROGRAM DATA:

- 1: Variable codes may not exceed an assigned value of 1500
- 2: Variable codes may not be assigned a value less than 6.
- 3: The number of NEW STAND cards in a single input stream may not exceed 10,000
- 4: The number of stand documentation cards per stand may not exceed 1,000
- 5: The number of inventory entries per stand may not exceed 10,000.

3.1.5. Program Error Messages

PROGRAM DATA will detect 2 types of input error types and print diagnostics back to the lineprinter. The 2 error messages are illustrated in Figure 3.13. along with the input stream that caused them.

Fatal Error No. 1 occurs whenever the first input image is not an ADD or CREATE command.

Fatal Error No. 2 occurs whenever a variable code that is not in the Variable Index File is read from input.

Any fatal errors detected by PROGRAM DATA will affect the following actions:

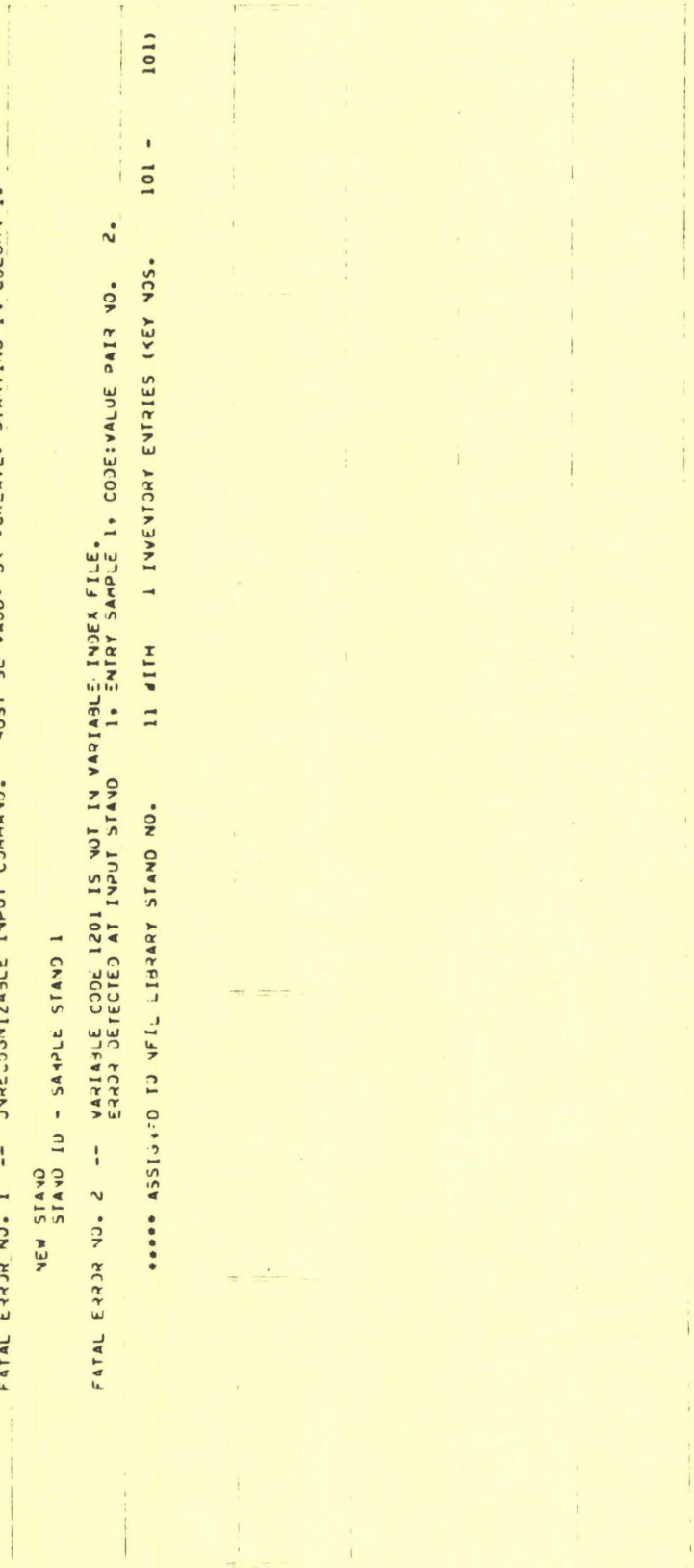
- the remaining input card images will be checked for correct form
- none of the new inventory data will be added to the Stand Documentation or Inventory Data files
- the existing Stand Documentation and Inventory Data files will be restored

Perhaps of greater importance are three input errors that will not be detected by PROGRAM DATA. PROGRAM DATA determines whether an input card image is a NEW STAND command, stand documentation image, or inventory entry documentation or code:value image by counting the number of blank columns preceeding the first data field on each image. Basicly:

- if columns 1 - 4 are not blank, a NEW STAND card is assumed, and no syntax check is made
- if columns 1 - 4 are blank, and columns 5 - 8 are not blank, the image is assumed to be a stand documentation card
- if columns 1 - 8 are blank, an inventory entry documentation is assumed, and columns 9 - 12 will be read for the number of trailing code:value pair images

It is the Library user's responsibility to ensure the input rules are satisfied. If undiagnosed errors result, it may be a result having data in the wrong columns.

FIGURE 3.13. PROGRAM DATA ERROR MESSAGES



3.2.6. Program Structure and Logic

PROGRAM DATA consists of a MAIN routine, 7 subroutines, and a BLOCK DATA routine. One labeled common block /ALL/ is used.

3.2.6.1. MAIN

- Step 1: Call SUBROUTINE HEADER to write output title page.
- Step 2: Read first input card image to determine whether a CREATE or ADD is requested.
- Step 3: If the first input card image is not CREATE or ADD, print an error diagnostic.
- Step 4: If CREATE, call SUBROUTINE CREATE to write a new file label (record type I) and file size (record type II) for new Stand Documentation and Inventory Data files.
- Step 5: Call SUBROUTINE PREP to read and store variable types and codes, and to read the Stand Documentation and Inventory Data files' labels and file size records.
- Step 6: Call SUBROUTINE NEWSTD to read and check the remaining input stream.
- Step 7: Call SUBROUTINE SMERGE to add new stand documentation records to the Stand Documentation File.
- Step 8: Call SUBROUTINE DMERGE to add new inventory entries to the Inventory Data File.
- Step 9: Call SUBROUTINE CLOSER to write closing output page.
- Step 10: STOP.

3.2.6.2. SUBROUTINE HEADER

- Step 1: Write PROGRAM DATA heading on output title page.
- Step 2: Write date and time PROGRAM DATA was submitted for execution on output title page.
- Step 3: RETURN to MAIN.

3.2.6.3. SUBROUTINE CREATE

- Step 1: Write new Stand Documentation File label (record type I) and file size (record type II) to logical device N2.
- Step 2: Write new Inventory Data File label (record type I) and file size (record type II) to logical device N3.
- Step 3: Write output message indicating that new Stand Documentation and Inventory Data files have been created.
- Step 4: REWIND logical devices N2 and N3.
- Step 5: RETURN to MAIN.

3.2.6.4. SUBROUTINE PREP

- Step 1: Read Variable Index File label (record type I) and file size (record type II) from logical device N1.
- Step 2: Read each variable and variable class and determine if the variable is continuous or not. If continuous, KODE2(KODE) = .TRUE. , where KODE is the variable code.
- Step 3: Read and store Stand Documentation File label (record type I) and file size (record type II) from logical device N2.
- Step 4: Read and store Inventory Data File label (record type I) and file size (record type II) from logical device N3.
- Step 5: Initialize the stand counter IDSTND to the number of stands currently on the Stand Documentation file, NSTAND (file size, record type II).
- Step 6: Initialize the inventory entry counter KEY to the number of inventory entries currently on the Inventory Data File, NKEYS (file size, record type II).
- Step 7: RETURN to MAIN.

3.2.6.5. SUBROUTINE NEWSTD

- Step 1: Read the next NEW STAND card.
- Step 2: Read each stand documentation card. If columns 1 - 4 are blank and columns 5 - 8 are not blank, a stand documentation card is assumed. If columns 1 - 8 are blank, an inventory entry documentation card is assumed, and control is passed to Step 3. The stand documentation card is written back to the lineprinter (logical device N5) and to logical device N8.
- Step 3: The inventory entry documentation card is read for the number of trailing code:value pairs, number of plots represented by the entry, and an 8 byte alphameric description. The entry is assigned the next sequential Library KEY number.
- Step 4: The inventory entry's KEY, number of plots, and alphameric description are written to logical device N9.
- Step 5: The inventory entry's number of trailing code:value pairs is incremented by 3 due to the addition of the 3 Library-assigned code:value pairs. The entry's KEY and NPAIRS is written to logical device N4.
- Step 6: The inventory entry's KEY, stand number, and number of plots it represents are assigned to variable codes 0001, 0002, and 0003, respectively. The three code:value pairs are then written to logical device N4.
- Step 7: The inventory entry's remaining code:value pairs are read from the input stream (logical device N5) and written to logical device N4. If a variable code that is not currently in the Variable Index File is read, an error diagnostic is printed.
- Step 8: The next input card image is read. If columns 1 - 8 are blank, a new inventory entry for the same stand is assumed, and control is passed to Step 3. If columns 1 - 4 are not blank, a NEW STAND is assumed, the old stand's Library-assigned stand number, number of inventory entries, KEY of the first stand inventory entry, and number of stand documentation cards are written to logical device N7 and to the lineprinter, and control is passed to Step 1.
- Step 9: If the end of the input file is detected, end of file marks are written to logical devices N4, N7, N8, and N9.
- Step 10: RETURN to MAIN.

3.2.6.6. SUBROUTINE SMERGE

- Step 1: Write the Stand Documentation File label (record type I) to logical device N10.
- Step 2: Write the file size (record type II) to logical device N10. If fatal errors were detected, the old file size is written. If no fatal errors occurred, the file size is set equal to the number of stands already on the Stand Documentation File plus the number of new stands processed by PROGRAM DATA.
- Step 3: The existing Stand Documentation File is read from logical device N2 and written onto logical device N10.
- Step 4: If no fatal errors were detected, the new stand documentation records are written onto logical device N10. Record type III (stand identification) is read from logical device N7, record type IV (stand description) is read from logical device N8, and record type V (inventory entry documentation) is read from logical device N9.
- Step 5: An end of file mark is written onto logical device N10.
- Step 6: RETURN to MAIN.

3.2.6.7. SUBROUTINE DMERGE

- Step 1: Write the Inventory Data File label (record type I) to logical device N11.
- Step 2: Write the Inventory Data File size (record type II) to logical device N11. If fatal errors were detected, the old file size is written. If no fatal errors occurred, the file size is set equal to the number of existing inventory entries plus the number of new inventory entries processed by PROGRAM DATA.
- Step 3: The existing Inventory Data File is read from logical device N3 and written to logical device N11.
- Step 4: If no fatal errors were detected, the new inventory data are written onto logical device N11. Record type III (inventory entry key:size pairs) are read from logical device N4, as are record type IV (inventory entry code: value pairs).
- Step 5: An end of file mark is written onto logical device N11.
- Step 6: RETURN to MAIN.

3.2.6.8. SUBROUTINE CLOSER

- Step 1: An output message is written indicating whether or not PROGRAM DATA terminated successfully and the number of fatal errors detected.
- Step 2: The number of new stands and their Library-assigned stand numbers is written to the lineprinter.
- Step 3: The number of new inventory entries and their Library-assigned keys is written to the lineprinter.
- Step 4: The date and time PROGRAM DATA terminated is printed.
- Step 5: RETURN to MAIN.

3.2.7. Logical Device Assignment

PROGRAM DATA uses 11 logical devices assigned to variables N1, N2, N3, N4, N5, N6, N7, N8, N9, N10, and N11. Logical devices are assigned to the variables in the BLOCK DATA routine.

DEVICE	USE
N1	Read-access to existing Variable Index File.
N2	Read-access to existing Stand Documentation File for ADD mode Write-access to create new Stand Documentation File for CREATE mode
N3	Read-access to existing Inventory Data File for ADD mode Write-access to create new Inventory Data File for CREATE mode
N4	Contains new Inventory Data File record types III and IV (inventory entry key:size and code:value pairs) processed from input
N5	Input
N6	Output
N7	Contains new Stand Documentation File record type III's (stand identification) processed from input
N8	Contains new Stand Documentation File record type IV's (stand descriptions) processed from input
N9	Contains new Stand Documentation File record type V's (inventory entry identification) processed from input
N10	Contains new Stand Documentation File at termination of PROGRAM DATA
N11	Contains new Inventory Data File at termination of PROGRAM DATA

3.2.7.1. Lawrence Berkeley Laboratory CDC 6000/7000

Logical devices are assigned in BLOCK DATA as follows:

```
      DATA N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11/1,2,3,4,5,6,7,8,9,
     + 10,11/
```

3.2.7.2. Fort Collins Computer Center UNIVAC 1100/82

Logical devices are assigned in BLOCK DATA as follows:

```
      DATA N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11/12,2,3,4,5,6,7,8,9,
     + 10,11/
```

3.3. PROGRAM

COUNT

3.3. PROGRAM COUNT

3.3.1. Purpose

PROGRAM COUNT is an NFIL file maintenance routine. It counts the occurrence of every variable code and variable class code in the Inventory Data File and enters the totals into the Variable Index File. The totals provide the Library user with a convenient reference to the type and amount of inventory information currently available in the Fuels Inventory Database by merely listing the Variable Index File.

3.3.2. Program Input

PROGRAM COUNT does not read card image input. All program input is from the Inventory Data File.

3.3.3. Program Output

PROGRAM COUNT does not generate any lineprinter output. The variable and variable class counts are entered into columns 17 - 24 of the Variable Index File's Variable Documentation records (record type III).

3.3.4. Program Limitations

3.3.4.1. Lawrence Berkeley Laboratory CDC 6000/7000

The LBL CDC version of PROGRAM COUNT employs the system's sort/merge utility to order every code:value pair in the Inventory Data File. This is necessary to overcome computer disk storage limitations which prevent the use of a double subscript counter array. Such an array would require approximately 500 x 10000 elements.

The LBL sort/merge facility also has limitations on the number of code:value pairs it can efficiently order. PROGRAM COUNT should not be used for databases with over 5000 inventory entries on file.

3.3.4.2. Fort Collins Computer Center UNIVAC 1100/82

The FCCC UNIVAC 1100 version of PROGRAM COUNT has the following limitations:

- 1: A maximum of 200 variables may be defined in the Variable Index File
- 2: No class of an ordinal variable may have a variable class code greater than 999

3.3.5. Program Error Messages

PROGRAM COUNT does not write any error messages since all input is from the Inventory Data File and all output is written onto the Variable Index File.

3.3.6. Program Structure and Logic

3.3.6.1. Lawrence Berkeley Laboratory CDC 6000/7000

The LBL CDC 6000/7000 version of PROGRAM COUNT consists of a main routine with no subroutines, functions, or block data routines.

- Step 1: Read the Variable Index File label and file size.
- Step 2: Read each variable and variable class record and re-assign variable codes to their sequence number within the file.
- Step 3: Read the Inventory Data File label and file size.
- Step 4: Read all Inventory Data File code:value pairs and write them to logical device N4.
- Step 5: Sort all code:value pairs on logical device N4 in ascending order and write the ordered list onto logical device N5.
- Step 6: Write new Variable Index File label and size records onto logical device N2.
- Step 7: Read and count occurrence of variable class codes for the first variable on the sorted code:value list from logical device N4.
- Step 8: After the first variable's classes have been counted, write its Variable Index File records, complete with occurrence counts, to logical device N2.
- Step 9: Go to Step 7 until all variables' classes have been read, counted, and written onto logical device N2.
- Step 10: STOP.

3.3.6.2. Fort Collins Computer Center UNIVAC 1100/82

The FCCC UNIVAC 1100 version of PROGRAM COUNT consists of a main routine with no subroutines, common blocks, functions, or block data routines.

- Step 1: Read the Variable Index File Label and file size.
- Step 2: Read each variable and variable class record and reassign variable codes to their sequence number within the file.
- Step 3: Read the Inventory Data File label and size.
- Step 4: Read all Inventory Data File code:value pairs, counting their occurrence in a double subscript array NC(I,J), where I is the variable's sequence number and J is the variable class.
- Step 5: Rewind the Variable Index File on logical device N1, and read and write the file label and size to logical device N2.
- Step 6: Read the remaining variable and variable class records from logical device N1, and write them, complete with occurrence counts, onto logical device N2.
- Step 7: STOP.

3.3.7. Logical Device Assignment

PROGRAM COUNT uses 6 logical devices for the LBL CDC 6000/7000 version, and 4 logical devices for the FCCC UNIVAC 1100 version. Logical devices are assigned to variables N1 through N6 in a DATA statement.

DEVICE	USE
N1	Read-access to the Variable Index File
N2	Write-access to the newly COUNTed Variable Index File
N3	Read-access to the Inventory Data File
N4	Contains unsorted code:value pairs for input to the LBL CDC 6000/7000 sort/merge utility
N5	Contains sorted code:value pairs output from the LBL CDC 6000/7000 sort/merge utility
N6	Output

3.3.7.1. Lawrence Berkeley Laboratory CDC 6000/7000

Logical devices are assigned as follows:

```
DATA N1,N2,N3,N4,N5,N6/1,2,3,4,5,6/
```

3.3.7.2. Fort Collins Computer Center UNIVAC 1100/82

Logical devices are assigned as follows:

```
DATA N1,N2,N3,N4,N5,N6/7,2,3,4,5,6/
```

3.4. PROGRAM LIST

3.4. PROGRAM LIST

3.4.1. Purpose

PROGRAM LIST is an NFIL file retrieval routine. It is available to Library users to obtain complete or partial listings of the Inventory Data File, Stand Documentation File, and/or Variable Index File.

3.4.2. Program Input

PROGRAM LIST input includes a 'file command' indicating the file to be listed, followed by a 'specification list' indicating the portion of the file to be printed. If no specification list is supplied, the entire file will be listed.

Any number of file commands may appear in a single PROGRAM LIST input stream, each followed by any number of specifications.

Figure 3.14. is a sample input stream to PROGRAM LIST. In line 1, the Library user has requested a listing of the Variable Index File. The specification list (lines 2 through 4) specifies variable codes 0001 - 0003, 0201, and 0401 - 441 to be listed.

Line 5 requests a listing of the Stand Documentation File. The list will be limited to stand numbers 1 - 2, 5, and 9 - 12.

Line 9 also requests a listing of the Stand Documentation File. It differs from the previous request in that all inventory entry documentation for the stands will be included in the listing. Stands 1 - 2, 4, and 10 - 9999999 will be listed. If fewer than 9999999 stands are on file, the listing will terminate with the last recorded stand.

Line 13 requests a partial listing of the Inventory Data File. The listing will include those inventory entries with key numbers 1 - 3, 50, and 98 - 9999999. Again, if fewer than 9999999 entries are on file, the listing will terminate with the last recorded stand.

The output generated by Figure 3.14. input stream is illustrated in Figures 3.15. through 3.19.

FIGURE 3.14. PROGRAM LIST SAMPLE INPUT STREAM

```
LINE                1         2         3         4         5         6
 NO   12345678901234567890123456789012345678901234567890123456789012345678901
  1   INDEX
  2                 1         3
  3               201
  4               411       441
  5   STANDS
  6                 1         2
  7                 5
  8                 9        12
  9   STANDS AND P_OTS
 10                 1         2
 11                 4
 12                10 99999999
 13   DATA
 14                 1         3
 15                50
 16                98 99999999
```

3.4.2.1. File Commands

PROGRAM LIST recognizes 4 file commands. Each command must begin in column 1 of the input card image, and must appear immediately ahead of its specification list (if one is supplied). The commands may appear in any order or any number of times desired by the Library user.

INDEX

The INDEX file command is used to obtain a listing of the Variable Index File (Figure 3.14., line 1).

STANDS

The STANDS file command is used to obtain a listing of the Stand Documentation File. The listing will include the stand number, number of inventory entries assigned it, the first and last entrys' key numbers, and all stand documentation.

STANDS AND PLOTS

The STANDS AND PLOTS file command (line 9) also requests a listing of the Stand Documentation File. It differs from the STANDS file command by including the key number, alphameric description, and number of plots represented by each inventory entry assigned to the stand.

DATA

The DATA file command (line 13) is used to obtain a listing of the Inventory Data File. The listing includes each inventory entry's key number, number of code:value pairs, and code:value pair list.

3.4.2.2. Specification Lists

Each file command may or may not be followed by a specification list. If no specification list is supplied, the entire file will be listed. If a specification list is supplied, only those file records explicitly or implicitly declared will be listed.

The specification list for a file command may consist of any number of card images. Each card image has the following format:

- columns 1 - 4 must be blank
- columns 5 - 12 contain the id of the first contiguous file record to be listed (right-justified)
- columns 13 - 20 contain the id of the last contiguous file record to be listed (right justified)

In Figure 3.14., line 2, Variable Index File variables 0001 through 0003 will be listed. Variables 0001 and 0003 were explicitly declared on the specification image, and variable 0002 was implicitly declared since it is located between variable 0001 and 0003.

If a single record is to be listed, its id should appear right-justified in columns 5 - 12, and columns 13 - 20 may be left blank. Line 3, for example, specifies variable 0201 to be listed.

An arbitrarily large id may be used in columns 13 - 20 of the last specification image to make sure all the records up to the end of file mark will be listed. This convention is illustrated in lines 12 and 16.

Specification list record id's are dependent upon the file to be listed:

- INDEX specification list id's are inventory variable codes as documented in the Variable Index file. All variable and variable classes with the specified code will be listed.
- STANDS and STANDS AND PLOTS specification id's are the Library-assigned stand numbers.
- DATA specification list id's are the Library-assigned inventory entry keys.

3.4.3. Program Output

PROGRAM LIST output generated by the sample input stream in Figure 3.14. is illustrated in Figures 3.15. through 3.19.

PROGRAM LIST first prints a title page showing the date and time the program was submitted for execution (Figure 3.15.).

Each file command and its specification list (if one is supplied) are echoed back to the lineprinter on a separate page. It is followed by the file's listing.

A description of the output generated by each file command may be found on the following pages. All file listings are reformatted to increase readability and conserve paper.

FIGURE 3.15. PROGRAM LIST SAMPLE TITLE PAGE OUTPUT

```
NATIONAL FUELS INVENTORY LIBRARY

**** PROGRAM  LIST ****
VERSION 01
LAST REVISED ON JUN 22, 1979

DEVELOPED BY
SYSTEMS FOR ENVIRONMENTAL MGMT
P.O. BOX 3776
MISSOULA, MONTANA 59806

FOR THE
NATIONAL FUELS INVENTORY AND APPRAISAL PROJECT
ROCKY MOUNTAIN FOREST AND RANGE EXPERIMENT STATION
FORT COLLINS, COLORADO

BEGIN ON
DATE :  28 SEP 79
TIME :  22.45.39
```

3.4.3.1. INDEX File Command Output

PROGRAM LIST output generated by the INDEX file command and specification list of Figure 3.14. (lines 1 through 4) appears in Figure 3.16.

The Variable Index File label is printed at the top of the first listing page followed by the number of inventory variable codes currently on the file.

The INDEX specification list requested variable codes 0001 through 0003, 0201, and 0411 through 0441 to be listed. It has resulted in a listing of 8 variables and their classes. Variables 0002, 0421, and 0431 are included because they were implicitly declared on the specification list.

The variable code, variable description, number of assigned classes, base 10 exponent multiplication factor, and number of inventory entries on the Inventory Data File referencing the code is printed. It is followed by a similar description of all the variable's classes.

The listing may be referenced by Library users to prepare input to PROGRAMs INDEX and DATA, or to determine the number of inventory entries on the Inventory Data File referencing each variable and variable class. For example, variable 0201, FOREST COVER TYPE, is referenced by 100 inventory entries; lodgepole pine has 40 entries, Douglas-fir has 50 entries, and western red cedar has the remaining 10 entries.

When preparing input to PROGRAMs INDEX or DATA, the Variable Index File list will indicate which codes have been assigned, continuous variable units of measure, and of great importance, the base 10 conversion factor used to convert continuous data to integer values required by PROGRAM DATA.

FIGURE 3.16. PROGRAM LIST 'INDEX' COMMAND SAMPLE OUTPUT

```
NATL FUELS INVENTORY LIBRARY -- VARIABLE INDEX FILE --  CREATED  22 SEP 79

NUMBER OF FUEL INVENTORY VARIABLE CODES NOW ON THE INDEX FILE IS      22

  1  NFIL INVENTORY ENTRY KEY                    1 CLASSES   1.0E 0 MULT. FACTOR      100 INVENTORY ENTRIES
     9999  SEQUENTIALLY-ASSIGNED 8 DIGIT INTEGER   CLASS NO.  1                      100 INVENTORY ENTRIES

  2  NFIL STAND NUMBER                           1 CLASSES   1.0E 0 MULT. FACTOR      100 INVENTORY ENTRIES
     9999  SEQUENTIALLY-ASSIGNED 8 DIGIT INTEGER   CLASS NO.  1                      100 INVENTORY ENTRIES

  3  NO. OF PLOTS REPRESENTED BY INVENTORY ENTRY  1 CLASSES   1.0E 0 MULT. FACTOR      100 INVENTORY ENTRIES
     9999  4 DIGIT INTEGER                         CLASS NO.  1                      100 INVENTORY ENTRIES

201  FOREST COVER TYPE (BY PLURALITY OF SPECIES)  3 CLASSES   1.0E 0 MULT. FACTOR      100 INVENTORY ENTRIES
      108  LODGEPOLE PINE                          CLASS NO.  1                       40 INVENTORY ENTRIES
      202  DOUGLAS-FIR                             CLASS NO.  2                       50 INVENTORY ENTRIES
      242  WESTERN RED CEDAR                       CLASS NO.  3                       10 INVENTORY ENTRIES

411  0.25 - 1.0 INCH DEAD & DOWN FUEL LOAD        1 CLASSES   1.0E-3 MULT. FACTOR      100 INVENTORY ENTRIES
     9999  TONS/ACRE                               CLASS NO.  1                      100 INVENTORY ENTRIES

421  1.0 - 3.0 INCH DEAD AND DOWN FUEL LOAD       1 CLASSES   1.0E-3 MULT. FACTOR      100 INVENTORY ENTRIES
     9999  TONS/ACRE                               CLASS NO.  1                      100 INVENTORY ENTRIES

431  > 3.0 INCH DEAD AND DOWN FUEL LOAD           1 CLASSES   1.0E-3 MULT. FACTOR      100 INVENTORY ENTRIES
     9999  TONS/ACRE                               CLASS NO.  1                      100 INVENTORY ENTRIES

441  TOTAL > 0.25 INCH DEAD AND DOWN FUEL LOAD    1 CLASSES   1.0E-3 MULT. FACTOR      100 INVENTORY ENTRIES
     9999  TONS/ACRE                               CLASS NO.  1                      100 INVENTORY ENTRIES
```

3.4.3.2. STANDS File Command Output

PROGRAM LIST output generated by the STANDS command and specification list of Figure 3.14. (lines 5 - 8) appears in Figure 3.17.

The Stand Documentation File label is printed across the top of the first listing page followed by the number of inventory stands currently on the file.

The STANDS specification list requested stands numbered 1 through 2, 5, and 9 through 12 to be listed. It has resulted in the listing of 5 stands and their documentation.

The stand number, number of inventory entries assigned to the stand, and the inventory entry keys for the first and last stand entry are printed on the first line of each stand's documentation. It is followed by the stand documentation cards supplied by the Library user via PROGRAM DATA.

The listing may be referenced by Library users to review the amount and type of inventory information currently available on the Inventory Data File. The listing also serves as a catalogue to Library-assigned stand numbers which may be used as input to PROGRAM REVIEW (see next section, 3.5.).

FIGURE 3.17. PROGRAM LIST 'STANDS' COMMAND SAMPLE OUTPUT

```
                NATL FUELS INVENTORY LIBRARY  -  STAND DOCUMENTATION FILE  -  CREATED  22 SEP 79

                        NUMBER OF FUEL INVENTORY STANDS CURRENTLY ON FILE IS       10

STAND NO.       1       10 INVENTORY ENTRIES ON FILE  (ENTRY NOS.       1 -      10)
                        STAND ID  - ALPHA
                        LOCATION  - BITTERROOT NF, SULA RD, SW MONTANA, USA
                        STAND     - SECOND GROWTH DOUGLAS-FIR AFTER 1940 CLEARCUT
                        SITE      - 3700 FT, NE ASPECT, HIGH PRODUCTIVITY
                        FUELS     - NATURAL
                        DATE      - JULY 1977

STAND NO.       2       10 INVENTORY ENTRIES ON FILE  (ENTRY NOS.      11 -      20)
                        STAND ID  - BETA
                        LOCATION  - BITTERROOT NF, WEST FORK RD, SW MONTANA, USA
                        STAND     - OLD GROWTH DOUGLAS-FIR REPLACING DECADENT LODGEPOLE STAND
                        SITE      - 5300 FT, EAST ASPECT
                        FUELS     - NATURAL
                        DATE      - AUGUST 1977

STAND NO.       5       10 INVENTORY ENTRIES ON FILE  (ENTRY NOS.      41 -      50)
                        STAND ID  - BETA
                        LOCATION  - LOLO NF, MISSOULA RD, WEST-CENTRAL MONTANA, USA
                        STAND     - LODGEPOLE PINE 80 YEARS AFTER WILDFIRE
                        SITE      - 6500 FT, SW ASPECT
                        FUELS     - NATURAL
                        DATE      - SEPTEMBER 1978

STAND NO.       9       10 INVENTORY ENTRIES ON FILE  (ENTRY NOS.      81 -      90)
                        STAND ID  - GAMMA
                        LOCATION  - CLEARWATER NF, KELLY CK RD, EAST-CENTRAL IDAHO, USA
                        STAND     - OLD GROWTH LODGEPOLE PINE ABOUT 225 YEARS AFTER WILDFIRE
                        SITE      - 4400 FT, SOUTH ASPECT
                        FUELS     - NATURAL
                        DATE      - JULY 1979

STAND NO.       10      10 INVENTORY ENTRIES ON FILE  (ENTRY NOS.      91 -     100)
                        STAND ID  - DELTA
                        LOCATION  - CLEARWATER NF, LOCHSA RD, EAST-CENTRAL IDAHO, USA
                        STAND     - OLD GROWTH WESTERN RED CEDAR
                        SITE      - 3200 FT, BOTTOM OF STEEP CANYON, WEST ASPECT
                        FUELS     - NATURAL
                        DATE      - SEPTEMBER 1979
```

3.4.3.3. STANDS AND PLOTS

PROGRAM LIST output generated by the STANDS AND PLOTS command and specification list of Figure 3.14. (lines 9 through 12) appears in Figure 3.18.

The Stand Documentation File label is printed across the top of the first listing page followed by the number of inventory stands currently on the file.

The STANDS AND PLOTS specification list requested stands numbered 1, 2, 4, and 10 through the end of file mark to be listed, resulting in the 4 stands listed in Figure 3.18. The stand number, number of inventory entries assigned, and keys of the first and last stand inventory entry are printed on the first line of each stand's listing. It is followed by the stand documentation cards supplied by the Library user via PROGRAM DATA.

The STANDS AND PLOTS listing differs from the STANDS listing by including each inventory entry's key, number of plots, and alphameric description. In Figure 3.18., the first inventory entry for stand number 10 has a key of 91, represents 1 inventory plot, and has an alphameric description 'DELTA 1'.

FIGURE 3.18. PROGRAM LIST 'STANDS AND PLOTS' COMMAND SAMPLE OUTPUT

```
NATL FUELS INVENTORY LIBRARY  -  STAND DOCUMENTATION FILE  -  CREATED  22 SEP 79

          NUMBER OF FUEL INVENTORY STANDS CURRENTLY ON FILE IS       10

STAND NO.      1      10 INVENTORY ENTRIES ON FILE  (ENTRY NOS.       1 -      10)
STAND ID  - ALPHA
LOCATION  - BITTERROOT NF, SULA RD, SW MONTANA, USA
STAND     - SECOND GROWTH DOUGLAS-FIR AFTER 1960 CLEARCUT
SITE      - 3700 FT, NE ASPECT, HIGH PRODUCTIVITY
FUELS     - NATURAL
DATE      - JULY 1977
        1   1 ALPHA  1          2   1 ALPHA  2          3   1 ALPHA  3          4   1 ALPHA  4
        5   1 ALPHA  5          6   1 ALPHA  6          7   1 ALPHA  7          8   1 ALPHA  8
        9   1 ALPHA  9         10   1 ALPHA 10

STAND NO.      2      10 INVENTORY ENTRIES ON FILE  (ENTRY NOS.      11 -      20)
STAND ID  - BETA
LOCATION  - BITTERROOT NF, WEST FORK RD, SW MONTANA, USA
STAND     - OLD GROWTH DOUGLAS-FIR REPLACING DECADENT LODGEPOLE STAND
SITE      - 5300 FT, EAST ASPECT
FUELS     - NATURAL
DATE      - AUGUST 1977
       11   1 BETA   1         12   1 BETA   2         13   1 BETA   3         14   1 BETA   4
       15   1 BETA   5         16   1 BETA   6         17   1 BETA   7         18   1 BETA   8
       19   1 BETA   9         20   1 BETA  10

STAND NO.      4      10 INVENTORY ENTRIES ON FILE  (ENTRY NOS.      31 -      40)
STAND ID  - ALPHA
LOCATION  - LOLO NF, SEELEY LAKE RD, WEST-CENTRAL MONTANA, USA
STAND     - OLD GROWTH DOUGLAS-FIR REPLACING DECADENT LODGEPOLE PINE
SITE      - 4500 FT, NW ASPECT
FUELS     - NATURAL
DATE      - JULY 1978
       31   1 ALPHA  1         32   1 ALPHA  2         33   1 ALPHA  3         34   1 ALPHA  4
       35   1 ALPHA  5         36   1 ALPHA  6         37   1 ALPHA  7         38   1 ALPHA  8
       39   1 ALPHA  9         40   1 ALPHA 10

STAND NO.     10      10 INVENTORY ENTRIES ON FILE  (ENTRY NOS.      91 -     100)
STAND ID  - DELTA
LOCATION  - CLEARWATER NF, LOCHSA RD, EAST-CENTRAL IDAHO, USA
STAND     - OLD GROWTH WESTERN RED CEDAR
SITE      - 3200 FT, BOTTOM OF STEEP CANYON, WEST ASPECT
FUELS     - NATURAL
DATE      - SEPTEMBER 1979
       91   1 DELTA  1         92   1 DELTA  2         93   1 DELTA  3         94   1 DELTA  4
       95   1 DELTA  5         96   1 DELTA  6         97   1 DELTA  7         98   1 DELTA  8
       99   1 DELTA  9        100   1 DELTA 10
```

3.4.3.4. DATA File Command Output

PROGRAM LIST output generated by the DATA command and specification list of Figure 3.14. (lines 13 through 16) appears in Figure 3.19.

The Inventory Data File label is printed across the top of the first listing page followed by the number of inventory entries currently on the file.

The DATA specification list requested inventory entries numbered 1 through 3, 50, and 98 through the end of file mark to be listed. It has resulted in the listing of 7 inventory entries and their code:value pairs.

The inventory entry key and its number of code:value pairs are printed as the first line of each inventory entry's list. It is followed by the entry's code:value pairs, with six pairs printed per line to reduce output volume. Inventory entry 50, for example, has 18 code:value pairs. The first pair has a code of 0001 and a value of 00000050, and the last pair has a code of 0801 and a value of 00126480.

The Library user is cautioned against indiscriminate use of the DATA file command without a restricted specification list. The most frequent use of the command will be to obtain the number of inventory entries on file (in which case the specification list should request entry number 1 only) or to check specific entries for suspect code:value pairs.

Use of the DATA command without a specification list will result in approximately N/10 pages of output, where N is the number of inventory entries on file.

FIGURE 3.19. PROGRAM LIST 'DATA' COMMAND SAMPLE OUTPUT

```
NFIL  -  INVENTORY DATA FILE  -  CREATED  22 SEP 79. LAST COUNTED ON

NUMBER OF FUEL INVENTORIES CURRENTLY ON THE DATA FILE IS      100

  1    18 VARIABLES
         1        1         2        1         3        1       101        3       111       26       201      202
       211       25       221        2       301     3780       411      449       421     2467       431   133644
       441   136560       501       12       601    13389       701        1       702    11119       801   147679

  2    18 VARIABLES
         1        2         2        1         3        1       101        3       111       26       201      202
       211       25       221        2       301     3984       411      535       421     3079       431    95124
       441    98739       501       14       601    14227       701        1       702    18614       801   117353

  3    18 VARIABLES
         1        3         2        1         3        1       101        3       111       26       201      202
       211       25       221        2       301     3437       411      489       421      965       431    58071
       441    59425       501        9       601     8289       701        1       702     9124       801    68549

 50    18 VARIABLES
         1       50         2        5         3        1       101       16       111       26       201      108
       211       10       221        1       301     6379       411      565       421     1426       431   113753
       441   115844       501       10       601    11521       701        1       702    10636       801   126480

 98    22 VARIABLES
         1       98         2       10         3        1       101        5       111       12       201      242
       211      400       221        1       301     3232       311        7       411      881       421     5212
       431    91031       441    87124       501       17       601    19748       701       11       702    65636
       711       33       721   111261       801   153760       901      110

 99    22 VARIABLES
         1       99         2       10         3        1       101        5       111       12       201      242
       211      400       221        1       301     3543       311        7       411     2610       421     2294
       431   286840       441   291744       501       11       601    24564       701       11       702    15687
       711       45       721    20432       801   308431       901       88

100    22 VARIABLES
         1      100         2       10         3        1       101        5       111       12       201      242
       211      400       221        1       301     3686       311        7       411     1955       421     8905
       431   106967       441   117827       501        6       601    99731       701       11       702    85577
       711       42       721   112268       801   203404       901      134
```

3.4.4. Program Limitations

3.4.4.1. Lawrence Berkeley Laboratory CDC 6000/7000

The following limitations are in effect for the LBL CDC 6000/7000 version of PROGRAM LIST:

- 1: Variable codes in the Variable Index File may not exceed an assigned value of 1500
- 2: A maximum of 100 specification images may follow each file command
- 3: There is no limit to the number of file commands used in the input stream

3.4.4.2. Fort Collins Computer Center UNIVAC 1100/82

The FCCC UNIVAC 1100 version of PROGRAM LIST has the same limitations as those listed above for the CDC 6000/7000.

3.1.5. Program Error Messages

PROGRAM LIST generates an error diagnostic if it is unable to interpret an input file command. In such cases, the unrecognizable command's specification list is read and ignored, and the next file command is interpreted.

3.4.6. Program Structure and Logic

PROGRAM LIST consists of a MAIN routine, 5 subroutines, a BLOCK DATA routine, and 1 labeled common block.

3.4.6.1. MAIN

- Step 1: Call SUBROUTINE HEADER to write output title page.
- Step 2: Read input file command from logical device N5.
- Step 3: If input file command is unrecognizable, print error diagnostic and dummy-read its specification list. Return to Step 2.
- Step 4: Read and decode file command specification list.
- Step 5: If INDEX file command, call SUBROUTINE INDEX.
- Step 6: If STANDS or STANDS AND PLOTS command, call SUBROUTINE STANDS.
- Step 7: If DATA file command, call SUBROUTINE DATA.
- Step 8: Return to Step 2 and read next file command.
- Step 9: STOP after encountering end of input file.

3.4.6.2. SUBROUTINE HEADER

- Step 1: Write PROGRAM LIST heading on output title page.
- Step 2: Write date and time PROGRAM LIST was submitted for execution to output title page.
- Step 3: RETURN to MAIN.

3.4.6.3. SUBROUTINE INDEX

- Step 1: Rewind Variable Index File on logical device N1 and reset file pointer to 1.
- Step 2: Read file label (record type I) and file size (record type II).
- Step 3: If a specification list was supplied with the INDEX file command, SUBROUTINE CODE is called to convert the specification list parameters from variable codes to variable sequence numbers as arranged in the Variable Index File.
- Step 4: The file pointer is positioned at the first specicification variable code.
- Step 5: The variable code record and its class records are read and output to the lineprinter.
- Step 6: Return to Step 4 and read next requested variable code.
- Step 7: RETURN to MAIN after reading and outputing last requested variable code.

3.4.6.4. SUBROUTINE CODE

- Step 1: Read the Variable Index File from logical device N1 and assign each variable code a sequence number according to its order on the file.
- Step 2: Re-assign specification list parameters from variable codes to variable code sequence numbers.
- Step 3: Rewind the Variable Index File and read the file label and file size records to return the pointer to its position as in SUBROUTINE INDEX.
- Step 4: RETURN to SUBROUTINE INDEX.

3.4.6.5. SUBROUTINE STANDS

- Step 1: Rewind the Stand Documentation File on logical device N2 and reset file pointer to position 1.
- Step 2: Read file label and file size.
- Step 3: Read first specification list parameter pair and position file pointer at requested stand.
- Step 4: Read the stand documentation records and write to the lineprinter. If STANDS AND PLOTS, write inventory entry documentation to lineprinter.
- Step 5: Return to Step 3 for next specification list parameter pair.
- Step 6: RETURN to MAIN.

3.4.6.6. SUBROUTINE DATA

- Step 1: Rewind the Inventory Data File on logical device N3 and reset file pointer to position 1.
- Step 2: Read file label and file size.
- Step 3: Read first specification list parameter pair and position file pointer at requested inventory entry.
- Step 4: Read inventory entry header and code:value pairs and write them to the lineprinter for all entries between the first and second specification parameters.
- Step 5: Return to Step 3 for next specification parameter pair.
- Step 6: RETURN to MAIN.

3.4.7. Logical Device Assignment

PROGRAM LIST uses 5 logical devices assigned to variables N1, N2, N3, N5, and N6. Logical devices are assigned to the variables in the BLOCK DATA routine.

DEVICE	USE
N1	Variable Index File
N2	Stand Documentation File
N3	Inventory Data File
N5	Input
N6	Output

3.4.7.1. Lawrence Berkeley Laboratory CDC 6000/7000

Logical devices are assigned in BLOCK DATA as follows:

```
DATA N1,N2,N3,N5,N6/1,2,3,5,6/
```

3.4.7.2. Fort Collins Computer Center UNIVAC 1100/82:

```
DATA N1,N2,N3,N5,N6/4,2,3,5,6/
```

3.5. PROGRAM REVIEW

3.5. PROGRAM REVIEW

3.5.1. Purpose

PROGRAM REVIEW is an NFIL file retrieval and summary routine. Its purpose is to count the occurrence of every variable code and variable class code for a user-specified set of inventory entries and print the results to the lineprinter. The absolute and percent occurrence of each variable code, absolute and percent occurrence of count and ordinal variable classes, and mean, standard deviation, minimum, and maximum values of continuous variables are output. The program provides the Library user with a means of reviewing the amount and type of inventory information available for a set of inventory entries. The selected entries are designated in the input stream utilizing the specification list convention as for PROGRAM LIST.

3.5.2. Program Input

PROGRAM REVIEW input consists of a specification parameter identifier followed by the specification list. The two allowable specification parameter identifiers are:

- STANDS
- PLOTS

The STANDS command identifies the specification list parameters as stand numbers. The PLOTS command identifies the specification list parameters as inventory entry keys.

The identifier must begin in column 1 of the input card image and appear immediately ahead of its specification list (Figure 3.20., line 1).

The trailing specification list must consist of at least 1 input card image indicating the STANDS or PLOTS to be REVIEWed. If no specification list is supplied, no review will be made.

The specification card image consists of two parameter fields. The first STAND or PLOT of a contiguous block to be reviewed appears right-justified in columns 5 through 12. The last STAND or PLOT of the contiguous block appears right-justified in columns 13 through 20. If only one isolated stand or plot is requested, its number should appear right-justified in the first parameter field, columns 5 through 12.

Figure 3.20. illustrates a sample PROGRAM REVIEW input stream. The review is to be based upon stand numbers (line 1), with stands 1 through 3, 5, and 9 through 12 to be reviewed as a single set.

To ensure that all STANDS or PLOTS up to the end of file mark will be included in a review, the second parameter field of the last specification image should contain an arbitrarily large number such as 9999999.

Consult Section 3.5.4. for limitations on the number of specification card input images per STANDS or PLOTS command, and the number of STANDS or PLOTS reviews permitted in a single input stream.

FIGURE 3.20. PROGRAM REVIEW SAMPLE INPUT STREAM

```
LINE          1         2         3         4         5         6         7         8
 NO   12345678901234567890123456789012345678901234567890123456789012345678901234567890
  1   STANDS
  2             1       3
  3             5
  4             9      12
```

3.5.3. Program Output

PROGRAM REVIEW output generated by the sample input stream of Figure 3.20. is illustrated in Figures 3.21. through 3.23.

A title page is written first including the date and time PROGRAM REVIEW was submitted for execution (Figure 3.21.). The first review input command and specification list is then read and echoed back to the lineprinter.

After reading the inventory entries for the selected STANDS or PLOTS, the results are printed on one or more pages as in Figure 3.22. The output is headed with the date and page number followed by a line indicating the number of inventory entries reviewed. In Figure 2.22., 60 inventory plots were reviewed for stands 1 through 3, 5, and 9 through 12 as requested on the specification list.

Next, each variable whose 4 digit code appeared at least once in the reviewed entries' code:value pairs lists has its code, alphameric description, absolute occurrence, and percent occurrence (relative to the number of reviewed entries) printed. Variable 0101, 'USDA NATIONAL FOREST', for example, was found on 60 of the reviewed entries, for an absolute occurrence of 100%. Variable 0311, 'ASPECT', was found in only 20 of the entries, for a 33.3% occurrence.

A line is then printed for every class of a reviewed variable. The class code, alphameric description, absolute occurrence, and percent occurrence (relative to the variable's absolute occurrence) is included for ordinal variables. For continuous and count variables, the class code, units of measure, mean, standard deviation, minimum, and maximum values are output.

Variable 0101, 'USDA NATIONAL FOREST', is ordinal and occurred in 60 of the 60 reviewed entries. Class 0003, 'BITTERROOT NATIONAL FOREST', accounted for 30 (50%) of the occurrences, and class 0005, 'CLEARWATER NATIONAL FOREST' accounted for another 20 (33.3%). For variable 0311, 'ASPECT', class 0005 'SOUTH' was found 10 times in the reviewed entries, for a 50% occurrence relative to the aspect code occurrence of 20.

Variable 0411, '0.25 - 1.0 INCH DEAD AND DOWN FUEL LOAD' is continuous and therefore has a single class (9999) describing the units of measure, 'TONS/ACRE'. It was recorded on all 60 of the reviewed entries with a mean of 3.995 tons per acre, standard deviation of 3.5176 tons per acre, and minimum and maximum values of 0.427 and 13.243 tons per acre, respectively.

After all counted variables and their classes have been printed, a closing page is written indicating the number of reviews made and the date and time PROGRAM REVIEW terminated (Figure 2.23.).

FIGURE 3.21. PROGRAM REVIEW SAMPLE TITLE PAGE OUTPUT

NATIONAL FUELS INVENTORY LIBRARY

**** PROGRAM REVIEW ****
VERSION 01
LAST REVISED ON JUL 07, 1979

DEVELOPED BY
SYSTEMS FOR ENVIRONMENTAL MGMT
P.O. BOX 3776
MISSOULA, MONTANA 59806

FOR THE
NATIONAL FUELS INVENTORY AND APPRAISAL PROJECT
ROCKY MOUNTAIN FOREST AND RANGE EXPERIMENT STATION
FORT COLLINS, COLORADO

BEGIN ON
DATE : 30 SEP 79
TIME : 19.23.48

30 SEP 79 PROGRAM REVIEW PAGE 1

TOTAL NUMBER OF INVENTORY RECORDS REVIEWED = 60

CODE	CLASS	DESCRIPTION	OCCURRED	(%)	MEAN	ST DEV	MINIMUM	MAXIMUM
1	0	NFIL INVENTORY ENTRY KEY	60	100.0				
	9999	SEQUENTIALLY-ASSIGNED 5 DIGIT INTEGER			45.5	34.6	1.0	100.0
2	0	NFIL STAND NUMBER	60	100.0				
	9999	SEQUENTIALLY-ASSIGNED 5 DIGIT INTEGER			5.0	3.4	1.0	10.0
3	0	NO. OF PLOTS REPRESENTED BY INVENTORY ENTRY	60	100.0				
	9999	4 DIGIT INTEGER			1.0	0.	1.0	1.0
101	0	USDA NATIONAL FOREST	60	100.0				
	3	BITTERROOT NATIONAL FOREST	30.	50.0				
	5	CLEARWATER NATIONAL FOREST	20.	33.3				
	16	LOLO NATIONAL FOREST	10.	16.7				
111	0	STATE	60	100.0				
	12	IDAHO	20.	33.3				
	25	MONTANA	40.	66.7				
201	0	FOREST COVER TYPE (BY PLURALITY OF SPECIES)	60	100.0				
	108	LODGEPOLE PINE	30.	50.0				
	202	DOUGLAS-FIR	20.	33.3				
	242	WESTERN RED CEDAR	10.	16.7				
211	0	STAND AGE	60	100.0				
	9999	YEARS			143.3	136.1	10.0	400.0
221	0	STAND MGMT STATUS (FUEL TREATMENT)	60	100.0				
	1	OLD GROWTH STAND	50.	83.3				
	2	SECOND GROWTH STOCKED STAND	10.	16.7				
	11	CLEAROUT LOGGING SLASH	0.	0.				
301	0	ELEVATION	60	100.0				
	9999	FEET ABOVE MEAN SEA LEVEL			5094.7	1559.0	2720.0	7852.0
311	0	ASPECT	20	33.3				
	1	NORTH	0.	0.				
	3	EAST	0.	0.				
	5	SOUTH	10.	50.0				
	7	WEST	10.	50.0				
	9	FLAT	0.	0.				
411	0	0.25 - 1.0 INCH DEAD + DOWN FUEL LOAD	60	100.0				
	9999	TONS/ACRE			1.0159	.5867	.3260	2.6120
421	0	1.0 - 3.0 INCH DEAD AND DOWN FUEL LOAD	60	100.0				
	9999	TONS/ACRE			3.9950	3.5176	.4270	13.2430
431	0	> 3.0 INCH DEAD AND DOWN FUEL LOAD	60	100.0				
	9999	TONS/ACRE			98.3925	61.9638	9.2770	290.4330
441	0	TOTAL > 0.25 INCH DEAD AND DOWN FUEL LOAD	60	100.0				
	9999	TONS/ACRE			103.4033	63.5058	10.4100	299.7610

FIGURE 3.22. PROGRAM REVIEW SAMPLE REVIEW PAGE OUTPUT

FIGURE 3.23. PROGRAM REVIEW SAMPLE CLOSING PAGE OUTPUT

```
**** PROGRAM REVIEW ****

1 REVIEWS WERE REQUESTED

CLOSE ON
19.23.50
30 SEP 79
```

3.5.4. Program Limitations

3.5.4.1. Lawrence Berkeley Laboratory CDC 6000/7000

The following limitations are in effect for the LBL CDC 6000/7000 version of PROGRAM REVIEW:

1: No variable codes may exceed a value of 1500

2: A maximum of 500 variable codes may be recorded in the Variable Index File

3: A maximum of 1000 specification images may follow the STANDS or PLOTS parameter identifier

4: Only 1 STANDS or PLOTS request may be made per PROGRAM REVIEW input stream

3.5.4.2. Fort Collins Computer Center UNIVAC 1100/82

The FCCC UNIVAC 1100 limitations on PROGRAM REVIEW are the same as limitation numbers 1, 2, and 3 for the CDC 6000/7000 above. Limitation number 4 does not apply to the UNIVAC 1100 version.

3.5.5. Program Error Messages

PROGRAM REVIEW will detect two error types and write descriptive diagnostics to the lineprinter.

If the specification list parameter identifier is neither STANDS nor PLOTS, an error message stating that the command was unrecognizable is printed and the trailing specification list is ignored.

If SUBROUTINE NEWKEY is unable to locate requested review STANDS numbers in the Stand Documentation File, the message

ERROR IN SUBR NEWKEY -- REQUESTED STAND (stand number) DOES NOT EQUAL POINTER STAND (stand number)

is printed on a separate page. It indicates that the file pointer location did not agree with the requested stand number, and either

- 1: the Stand Documentation File structure is in error, or
- 2: a bug exists in PROGRAM REVIEW

3.5.6. Program Structure and Logic

PROGRAM REVIEW consists of a MAIN routine, 9 subroutines, a BLOCK DATA routine, and 2 labeled common blocks.

The labeled common block /QCOMPILR/ is specific to the CDC version of PROGRAM REVIEW and is used to assign parameters to the sort/merge utility.

3.5.6.1. MAIN

- Step 1: Call SUBROUTINE HEADER to write title output page.
- Step 2: Call SUBROUTINE INDEX to read, convert, and store Variable Index File variable codes as sequence numbers.
- Step 3: Read and interpret specification list parameter identifier command.
- Step 4: If identifier command is neither PLOTS nor STANDS, the specification list is dummy-read and an error diagnostic printed. Control is passed to Step 3.
- Step 5: Zero-initialize the variable code counter array.
- Step 6: Call SUBROUTINE LIST to read the specification list.
- Step 7: If another REVIEW is requested, go to Step 3.
- Step 8: Call SUBROUTINE CLOSER to write closing output page.
- Step 9: STOP.

3.5.6.2. SUBROUTINE HEADER

- Step 1: Write PROGRAM REVIEW title on first output page.
- Step 2: Write date and time PROGRAM REVIEW was submitted for execution.
- Step 3: RETURN to MAIN.

3.5.6.3. SUBROUTINE INDEX

- Step 1: Read Variable Index File label and size.
- Step 2: Read variable code and reset it to its sequence number (order of appearance in the file).
- Step 3: Read each of the variable's classes.
- Step 4: If the class code equals 9999, variable is continuous and CONTIN(IVAR) = .TRUE.
- Step 5: Go to Step 2 and read next variable.
- Step 6: RETURN to MAIN.

3.5.6.4. SUBROUTINE LIST

- Step 1: Read the specification card input image and decode the first and second parameter values.
- Step 2: If the parameters are based on STANDS, call SUBROUTINE NEWKEY to re-assign the parameters to inventory entry keys.
- Step 3: Call SUBROUTINE TALLY to read and tally variable code occurrence in the Inventory Data File.
- Step 4: Go to Step 1 and read next specification card image.
- Step 5: If end of input file encountered, call SUBROUTINE STATS to compute descriptive and occurrence statistics.
- Step 6: RETURN to MAIN.

3.5.6.5. SUBROUTINE NEWKEY

- Step 1: Rewind Stand Documentation File, read file label and size, and reset pointer to 1.
- Step 2: Position pointer at first specification image stand number.
- Step 3: Read the stand's first inventory entry key and reset the first specification list parameter equal to the key.
- Step 4: Position pointer at second specification image stand number.
- Step 5: Read the stand's last inventory entry key and reset the second specification parameter equal to the key.
- Step 6: RETURN to SUBROUTINE LIST.

3.5.6.6. SUBROUTINE TALLY

- Step 1: If first call to SUBROUTINE TALLY, read Inventory Data File label and size and reset file pointer to 1.
- Step 2: If first or second specification parameter is beyond file size, reset it equal to the file size.
- Step 3: Position the file pointer at the first inventory entry to be reviewed.
- Step 4: Read the entry's key and number of code:value pairs.
- Step 5: Increment the inventory entry counter.
- Step 6: Read code:value pair and reset the variable code equal to the variable code sequence number.
- Step 7: Increment the variable code counter and write the code's sequence number and value to logical device N4.
- Step 8: Return to Step 6 and read the next code:value pair.
- Step 9: Return to Step 4 and read the next inventory entry.
- Step 10: RETURN to SUBROUTINE LIST.

3.5.6.7. SUBROUTINE STATS

- Step 1: Submit code:value pairs on logical device N4 to sort routine. Sorted code:value pairs are written onto logical device N7.
- Step 2: Read code:value pairs for first variable code.
- Step 3: If CONTIN(IVAR) = .FALSE., variable class counter is incremented for every variable class value read (X(KLASS) = X(KLASS) + 1)
- Step 4: If CONTIN(IVAR) = .TRUE., the value is converted to a floating point real number using the variable's base 10 exponent conversion factor. The sum of the variable value is stored in X(1), the sum of squares in X(2), the minimum in X(3), and the maximum in X(4).
- Step 5: Go to Step 4 and read next continuous variable value.
- Step 6: After all the variable's values have been read and stored, the mean, standard deviation, minimum, and maximum are computed.
- Step 7: SUBROUTINE PRINT1 is called to write variable and variable class occurrence output and/or descriptive statistics.
- Step 8: RETURN to SUBROUTINE LIST.

3.5.6.8. SUBROUTINE PRINT1

- Step 1: If first call to SUBROUTINE PRINT1, rewind Variable Index File on logical device N1 and read file label and size. Reset file pointer to position 1.
- Step 2: Position file pointer at variable whose code and class occurrence is to be printed.
- Step 3: If first line of a new output page, call SUBROUTINE PRINTH to print new page column headings.
- Step 4: Read and write the variable's code, description, absolute occurrence, and percent occurrence to the lineprinter.
- Step 5: If an ordinal variable, read and write each class's code, description, absolute, and percent occurrence to the lineprinter.
- Step 6: If a continuous variable, read and write the class's code, units of measure, mean, standard deviation, minimum, and maximum values to the lineprinter.
- Step 7: Go to Step 2 for next variable.
- Step 8: RETURN to SUBROUTINE LIST.

3.5.6.9. SUBROUTINE PRINTH

- Step 1: Write column headins as first line of new output page.
- Step 2: RETURN to SUBROUTINE PRINT1.

3.5.6.10. SUBROUTINE CLOSER

- Step 1: Write PROGRAM REVIEW closing title.
- Step 2: Write number of REVIEWS requested.
- Step 3: Write date and time PROGRAM REVIEW terminated.

3.5.7. Logical Device Assignment

PROGRAM REVIEW uses 7 logical devices assigned to variables N1, N2, N3, N4, N5, N6, and N7 in the BLOCK DATA routine.

DEVICE	USE
N1	Variable Index File
N2	Stand Documentation File
N3	Inventory Data File
N4	Unsorted reviewed code:value pairs
N5	Input
N6	Output
N7	Sorted code:value pairs

3.5.7.1. Lawrence Berkeley Laboratory CDC 6000/7000

The LBL CDC 6000/7000 version of PROGRAM REVIEW assigns logical devices in the BLOCK DATA routine as follows:

```
DATA N1,N2,N3,N4,N5,N6,N7/1,2,3,4,5,6,7/
```

3.5.7.2. Fort Collins Computer Center UNIVAC 1100/82

The FCCC UNIVAC 1100 version of PROGRAM REVIEW has the following logical device assignments in the BLOCK DATA routine:

```
DATA N1,N2,N3,N4,N5,N6,N7/8,2,3,4,5,6,7/
```

3.6. PROGRAM SEARCH

3.6.1. Purpose

PROGRAM SEARCH is an NFIL data retrieval, analysis, and summary routine. It permits the Library user to:

- search the Fuels Inventory Database and retrieve values of selected inventory variables. Variables whose values are to be retrieved are termed 'retrieval' variables. If the variable's values are to be printed in an output table, it may also be referred to as a 'table' variable.

  For example, the Library user may wish to retrieve total dead and down fuel load values from all inventory entries. 'TOTAL DEAD AND DOWN FUEL LOAD' would then be a retrieval or table variable.

- constrain the retrieved variable values to those inventory entries with different or the same variables whose values meet user-specified requirements. These are called 'constraint' variables.

  In the above example, the user may wish to constrain retrieved total dead and down fuel load to those inventory entries taken between 3000 and 6000 feet elevation. Elevation is then a constraint variable. The user may also wish to constrain total fuel load values retrieved to those less than 100 tons per acre in an attempt to eliminate 'jackpots' from the analysis. Thus, 'TOTAL DEAD AND DOWN FUEL LOAD' may be both a retrieval and a constraint variable.

- stratify the retrieval variables into classes of none, one, or two simultaneous 'stratification' variables.

  To continue the example, the user may wish to stratify the retrieved dead and down fuel loads into three cover type classes and 9 stand age classes. 'FOREST COVER TYPE' is therefore a stratification variable with classes of LODGEPOLE PINE, DOUGLAS-FIR, and WESTERN RED CEDAR. 'STAND AGE' is the second stratification variable with 9 user-defined age classes.

- create a temporary peripheral data file containing table variable values and their stratification class assignments.

- perform statistical analyses of the retrieval variable values within the specified stratification scheme. Descriptive statistics may be weighted, if desired, by the number of plots represented by each inventory entry. Descriptive statistics include:

  -- absolute occurrence (sample size)
  -- percent occurrence

-- sums
-- sums of squares
-- means
-- variances
-- standard deviations
-- standard errors
-- percent presence
-- minimum values
-- maximum values

3.6.2. Program Input

PROGRAM SEARCH recognizes 10 input commands. Commands may or may not include qualifiers, parameters, arguments, and/or specification lists. The commands and their forms are:

- RUN NAME parm1 parm2 ... parm16
- COMMENT parm1 parm2 .... parm18
- CONSTRAIN VAR parm1
  EXCLUDE IF arg1 parm2
- ROW qual kode
  parm1 parm2 parm3
- COL qual kode
  parm1 parm2 parm3
- TABLE VAR parm1
- OCCURRENCE arg1
  parm1
- STATISTICS arg1
  parm1
- SEARCH
- END

All input commands, qualifiers, arguements, and parameters are demonstrated in the sample input stream of Figure 3.24.

FIGURE 3.24. PROGRAM SEARCH SAMPLE INPUT STREAM

```
LINE        1         2         3         4         5         6         7         8
 NO 12345678901234567890123456789012345678901234567890123456789012345678901234567890
  1 RUN NAME          COVER TYPE VS STAND AGE
  2 COMMENT           USE ALL FOREST SERVICE DATA
  3 COMMENT           CONSTRAIN ELEVATION TO 3200 - 7400 FEET
  4 CONSTRAIN VAR      301
  5     EXCLUDE IF     LT    3200
  6     EXCLUDE IF     GT    7400
  7 COMMENT           COVER TYPE WILL BE THE ROW STRATIFICATION
  8 ROW LIST           201
  9                        108            LODGEPOLE PINE
 10                        202            DOUGLAS-FIR
 11                        242            WESTERN RED CEDAR
 12 COMMENT           STAND AGE WILL BE THE COLUMN STRATIFICATION
 13 COL RANGE         211
 14                         1      50      1- 50
 15                        51     100     51-100
 16                       101     200    101-200
 17                       201     250    201-250
 18                       301     500    301-500
 19 COMMENT           RETRIEVE 0.25 - 1.0 INCH FUEL LOAD
 20 TABLE VAR          411
 21 COMMENT           RETRIEVE 1.0 - 3.0 INCH FUEL LOAD
 22 TABLE VAR          421
 23 COMMENT           RETRIEVE GREATER THAN 3 INCH FUEL LOAD
 24 TABLE VAR          431
 25 COMMENT           RETRIEVE TOTAL DEAD AND DOWN FUEL LOAD
 26 TABLE VAR          441
 27 COMMENT           RETRIEVE DEAD AND DOWN FUEL INTERCEPT HEIGHT
 28 TABLE VAR          501
 29 COMMENT           RETRIEVE DEAD AND DOWN FUEL BULK DENSITY
 30 TABLE VAR          501
 31 COMMENT           RETRIEVE SHRUB COVER
 32 TABLE VAR          901
 33 COMMENT           DETERMINE SAMPLE SIZE, MEAN, AND ST DEV OF TABLE VARIABLES
 34 COMMENT           NO WEIGHTING BY ENTRY PLOT SIZE
 35 STATISTICS
 36                      1
 37                      4
 38                      6
 39 COMMENT           DETERMINE SAMPLE SIZES IF WEIGHTED BY ENTRY PLOT SIZE
 40 OCCURRENCE        WEIGHTED
 41                      1
 42 COMMENT           END OF RUN 1
 43 SEARCH
 44 COMMENT           ONLY 1 RUN DURING THIS INPUT STREAM
 45 END
```

3.6.2.1. RUN NAME Card

The RUN NAME card permits the Library user to specify a PROGRAM SEARCH run title that will appear at the top of every table printed for the run. The RUN NAME card has the following format:

RUN NAME parm1 parm2 parm3 ... parm16

The command RUN NAME must appear on the card beginning in column 1. The parameters parm1 through parm16 are 4 character alphameric run identifiers beginning in columns 17 and ending in column 80.

When PROGRAM SEARCH encounters a RUN NAME card, all information in columns 17 through 80 are read, stored, and printed as the title to every table produced by the run.

Note: information in columns 9 through 16 will be ignored.

The RUN NAME card may appear anywhere in a run stream (e.g., before a SEARCH card). If it appears more than once in a run stream, the information on the last card will be printed as table titles.

An example of the RUN NAME card appears in Figure 3.24., line 1.

3.6.2.2. COMMENT Card

The COMMENT card permits the Library user to insert clarifying statements within the PROGRAM SEARCH run stream. It has the same format and syntax as the RUN NAME card. Information on the COMMENT card will appear only in the input stream echo for the run (see Figure 3.27.). The COMMENT card has the following format:

COMMENT parm1 parm2 parm3 ... parm18

The command COMMENT must appear on the card beginning in column 1. The parameters parm1 through parm18 are 4 character alphameric information beginning in any column and ending in column 80.

When PROGRAM SEARCH encounters a COMMENT card, all information on the input card image is written directly back to the lineprinter and appears in the input stream echo.

A COMMENT card may appear anywhere in the run stream. Examples of COMMENT card use may be found in Figure 3.24., lines 2, 3, 7, 12, 19, 21, 23, 25, 27, 29, 31, 33, 34, 39, 42, and 44.

3.6.2.3. CONSTRAIN VAR Card

The CONSTRAIN VAR card is used to identify a constraint variable (see Section 3.1., Purpose). The command is followed by a parameter containing the constrained variable's 4 digit code. Following the CONSTRAIN VAR card is a specification list of one or more input card images containing the specification EXCLUDE IF. The EXCLUDE IF specification is followed by an arithmetic argument and an operand value.

The CONSTRAIN VAR card and its associated EXCLUDE IF specifications form a paragraph with the following structure:

```
         1         2         3         4
1234567890123456789012345678901234567890

CONSTRAIN VAR   KODE
    EXCLUDE IF    AA  KVALUE
    EXCLUDE IF    AA  KVALUE
    ....... ..    ..  ......
    EXCLUDE IF    AA  KVALUE
```

The CONSTRAIN VAR must begin in column 1, and all trailing EXCLUDE IF specifications must begin in column 5 (if columns 1 through 4 are not blank, PROGRAM SEARCH will attempt to interpret the field as a command).

KODE is the 4 digit constraint variable code, as per the Variable Index File, right-justified in columns 17 - 20 of the CONSTRAIN VAR card.

AA is an arithmetic argument. Allowable arguments are:

- LT = less than
- LE = less than or equal to
- EQ = equal to
- NE = not equal to
- GE = greater than or equal to
- GT = greater than

KVALUE is the argument operand completing the EXCLUDE IF specification. The KVALUE for continuous variables must be an integer value equal to

$$KVALUE = IFIX(V / 10**M)$$

where V is the floating point operand value and M is the variable's base 10 exponent conversion factor. For example, if fuel load is to be constrained to values less than 20.0 tons per acre, and fuel load conversion factors are -3, then KVALUE equals 20000. KVALUE must also be right-justified in columns 21 through 28. The arithmetic argument AA must be in columns 19 and 20.

A single CONSTRAIN VAR card may use as many EXCLUDE IF specifications as necessary to constrain retrieved variables to selected values.

The CONSTRAIN VAR and its EXCLUDE IF cards must appear as a contiguous paragraph in the input stream. Other commands may not be inserted into the paragraph.

Consult Section 3.6.4. for limitations on the number of CONSTRAIN VAR cards allowed per input run, and the number of EXCLUDE IF specifications permitted per CONSTRAIN VAR card.

In Figure 3.24., variable 0301, 'ELEVATION', has been defined as a constraint variable. No table variable values will be retrieved for an inventory entry whose elevation is less than 3200 feet (line 5) or greater than 7400 feet (line 6).

3.6.2.4. ROW and COL Cards

The ROW and COL commands are used to designate row and column stratification variables, respectively. Syntax is identical for both commands. Each command card is followed by one or more specification cards defining the classes of the stratification. The command form is:

```
         1         2         3         4         5         6
12345678901234567890123456789012345678901234567890123456789012345678901234567890

ROW qual        kode
                aaaaaaaabbbbbbbb    ccccccccccccccccccc
                aaaaaaaabbbbbbbb    ccccccccccccccccccc
```

The ROW (or COL) command must begin in column 1 of the first card of the paragraph.

'kode' is the stratification variable's 4 digit code as per the Variable Index File. It appears right-justified in columns 17 - 20.

'qual' is a ROW or COL qualifier indicating how the stratification classes are to be assigned in the trailing specification list. Three qualifiers are recognized:

- LIST
- GROUP
- RANGE

Each qualifier is discussed separately on the following pages. Fields aaaaaaaa and bbbbbbbb are stratification class definition parameters specific to the declared qualifier. The ccc...ccc field contains alphameric stratification class descriptions which are to be used as row or column headings in the output tables.

The ROW or COL card and its trailing specification list constitutes a contiguous paragraph that may not have other commands, such as a COMMENT card, inserted into it.

Consult Section 3.6.4. for restrictions on the number of specification images which may follow a ROW or COL card.

LIST

The LIST qualifier indicates that the trailing specification images will contain a list of variable classes which define the stratification classes; each variable class is the unique member of a stratification class. The LIST was designed primarily for use in defining stratification classes for ordinal variables.

In line 8, Figure 3.24., variable 0201 'FOREST COVER TYPE' is an ordinal variable designated as the ROW stratification. The stratification class LIST contains the forest cover type variable class codes right-justified in columns 17 through 24 (field aaaaaaaa above). Columns 25 through 32 are left blank (field bbbbbbbb). Columns 37 through 56 contain a 20 character row class description. The description will be used as row or column identifiers in the output tables.

Another example of the ROW and LIST paragraph syntax is:

```
         1         2         3         4         5         6
12345678901234567890123456789012345678901234567890123456789012345678901234567890

ROW LIST        0201                    SAF FOREST COVER TYPE
                  0101                  WHITEBARK PINE
                  0108                  LODGEPOLE PINE
                  0202                  DOUGLAS-FIR
                  0242                  WESTERN RED CEDAR
```

- The ROW (or COL) command must begin in column 1 of the first card.
- The LIST qualifier must begin in column 5 of the same card as ROW.
- The ROW or COL stratification variable code must be right-justified in columns 17 through 20 of the first card.
- Library user comments may appear anywhere on columns 21 through 80.
- The variable class to be designated as the first stratification class has its code right-justified in columns 17 through 24.
- The stratification class description appears in columns 37 - 56.

In the above example, four stratification classes have been defined for the stratification variable 'SAF FOREST COVER TYPE'. 'WHITEBARK PINE' (class 0101) will be assigned as the first stratification class, and 'WESTERN RED CEDAR' will be assigned as the fourth class. Other cover types not explicitly declared in the LIST will not have their table variables retrieved from the Inventory Data File.

GROUP

The GROUP qualifier is a variation on the LIST qualifier. It indicates that the variable class whose code appears in field aaaaaaaa (columns 17 - 24) will be assigned to a stratification class whose number appears in field bbbbbbbb (columns 25 - 32). In short, it permits the user to GROUP ordinal variable classes into a common stratification class. While designed primarily for use with ordinal variables, situations may arise where it can be used with count or continuous variables.

To illustrate use of the GROUP qualifier, refer to the example on the previous page using the LIST qualifier. Suppose whitebark pine is to be grouped with lodgepole pine in a single stratification class. This would be accomplished by:

```
         1         2         3         4         5         6
12345678901234567890123456789012345678901234567890123456789
```

ROW GROUP 0201 SAF FOREST COVER TYPE

0101	1	
0108	1	LODGEPOLE/WHITEBARK
0202	2	DOUGLAS-FIR
0242	3	WESTERN RED CEDAR

(Underlined columns are read input fields.)

'SAF FOREST COVER TYPE' (variable 201) is again the row stratification variable. Whitebark pine (variable class 101) is assigned to stratification class 1, as is lodgepole pine (variable class 108). Douglas-fir is assigned to stratification class 2, and western red cedar to stratification class 3.

Each time PROGRAM SEARCH reads a specification list, it stores the class description in columns 37 - 56 for later printing as row or column table headings. Therefore the last class description for a GROUP will be the one printed in the output tables. It was not necessary then to include a class description when whitebark pine (variable class 101) was assigned to group 1, but was necessary when lodgepole pine was assigned (e.g., lodgepole pine was the last group 1 assignment).

The variable class code must appear right-justified in columns 17 - 24, and the stratification class number must appear right-justified in columns 25 - 32. Consult Section 3.6.4. for restrictions on the number of variable class assignments and stratification classes allowed.

Note: variable classes not explicitly assigned to a stratification class will not have their associated table variables retrieved from the Inventory Data File by PROGRAM SEARCH.

RANGE

The RANGE qualifier indicates that the specification images will contain a lower stratification class boundary in field aaaaaaaa (columns 17 - 24) and an upper stratification class boundary in field bbbbbbbb (columns 25 - 32). The RANGE qualifier was designed primarily for use with continuous or count variables, but situations may arise where it can expidite ordinal variable class stratification assignment.

Each specification image following a RANGE qualifier defines a single stratification class whose boundaries are defined in the aaaaaaaa and bbbbbbbb field. The class includes the variable values defining the boundaries.

In Figure 3.24., lines 13 through 18, continuous variable 0211 'STAND AGE' has been designated the column stratification variables. The 9 trailing specification images define 9 stand age classes. The first class, 1 to 50 years, has the lower boundary in columns 17 - 24, and the upper boundary in columns 25 - 32. The class description appears in columns 37 - 56. Since the class includes the boundary values, the second class has '51' for a lower boundary value. All boundary values must be right-justified in their respective fields.

Important: All boundary values must be expressed as integers with the same number of significant digits as stored on the Inventory Data File. The conversion from floating point to integer values is made from the formula:

```
IBOUND = IFIX( XBOUND / (10**M) )
```

where

```
IBOUND = stratification class integer boundary
         value
XBOUND = stratification class floating point
         boundary value
M      = variable's base 10 exponent conversion
         factor as per the Variable Index File
```

In the example above and in Figure 3.24., the stratification variable 'STAND AGE' has an M of 0, and therefore IBOUND = IFIX(XBOUND).

To further illustrate the integer value requirement, suppose the column stratification variable is 'TOTAL FUEL LOAD' (variable code 441), and it is to have three classes; 0.000 tons per acres, 0.001 to 100.000 tons per acre, and greater than 100.000 tons per acre. The M for fuel load values is -3. The proper COL paragraph is:

```
         1         2         3         4         5         6
1234567890123456789012345678901234567890123456789012345678901234567890
```

COL RANGE	0441		TOTAL FUEL LOAD
	0000	0000	0.000 TONS/ACRE
	0001	100000	0.001 - 100.000 T/A
	100001	9999999	OVER 100.000 T/A

The first class, 0.0 tons per acre, has a range from 0.0 to 0.0. The second class has a range of

0.001 / (10**-3) = 1

to

100.000 / (10**-3) = 100000.

tons per acre. The third class has a range of

100.001 / (10**-3) = 100001

to

9999.999 / (10**-3) = 9999999

tons per acre.

Returning to the example in the GROUP qualifier discussion, whitebark pine and lodgepole pine may have been assigned to the same stratification class as follows:

```
         1         2         3         4         5         6
1234567890123456789012345678901234567890123456789012345678901234567890
```

ROW RANGE	0201		SAF FOREST COVER TYPE
	101	108	LODGEPOLE/WHITEBARK
	202	202	DOUGLAS-FIR
	242	242	WESTERN RED CEDAR

The above paragraph woul accomplish the same results as that in the GROUP discussion. However, any cover type classes with variable codes greater than 101 and less than 108 would also be included in the stratification class 'LODGEPOLE/WHITEBARK'.

3.6.2.5. TABLE VAR Card

The TABLE VAR command is used to identify a retrieval or table variable. Retrieval or table variables are those variables whose values are retrieved, summarized, presented in the output tables, and written to the peripheral data file. The TABLE VAR card has a single parameter and has the form:

```
         1         2
12345678901234567890
TABLE VAR      kode
```

The command TABLE VAR must begin in column 1. 'kode' is the variable's code as per the Variable Index File. The TABLE VAR command has no trailing specification list.

One TABLE VAR card is required for each retrieval variable. Figure 3.24. designates 7 retrieval variables at lines 20, 22, 24, 26, 28, 30, and 32.

Consult section 3.6.4. for limitations on the number of TABLE VAR cards permitted per run.

3.6.2.6. STATISTICS Card

The STATISTICS card is used to identify the type and amount of statistics to be computed for all table variables in the run. It has a single argument followed by 1 or more specification images. The form of the command is:

```
         1         2
12345678901234567890123
STATISTICS      aaaa
                bbbb
                bbbb
                ....
                bbbb
```

where aaaa is the weighting argument and bbbb is a statistic number. If aaaa is left blank, each inventory entry is considered a single sample (weighting factor of 1). If aaaa is non-blank, each inventory entry's retrieved variable values will be weighted by the number of plots represented by the entry. Therefore, if no weighting is requested, plot by plot and stand summary entries will all be treated as single samples. If weighting is specified, plot by plot entries will be single samples and stand summary entries will be weighted by the number of inventory plots they represent.

The bbbb field contains a right-justified statistic specification. The following statistics and their numbers are available:

1 sample size
2 sums
3 sums of squares
4 means
5 variances
6 standard deviations
7 standard errors
8 percent presence
9 minimum values
10 maximum values

In Figure 3.24., lines 35 through 38, statistics 1 (sample size), 4 (means), and 6 (standard deviations) have been requested.

From 1 to 10 statistics may be requested per run. Each statistic requested will result in a one or two page output table for each table variable. The number of tables produced per run is therefore the product of the number of table variables requested and the number of statistics specified.

3.6.2.7. OCCURRENCE Card

The OCCURRENCE command is used to identify the type and amount of table variable occurrence tabulation to perform for all the run table variables. Its use is preferred over the STATISTICS card for performing cross-tabulations of variable occurrences within a stratification scheme. The OCCURRENCE card has a single argument followed by 1 or more specification images. The form of the command is:

```
         1         2
12345678901234567890

OCCURRENCE      aaaa
                bbbb
                bbbb
                ....
                bbbb
```

where aaaa is the weighting argument and bbbb is an occurrence statistic number. If aaaa is left blank, each inventory entry is considered a single sample (weighting factor of 1). If non-blank, each entry's retrieved variable values will be weighted by the number of plots represented by the entry. In the latter case, all plot by plot inventory entries will have a weighting factor of 1 and all stand summary entries will have a weighting factor greater than 1.

The bbbb field contains a right-justified occurrence statistic in columns 17 through 20. The occurrence statistics are:

1 absolute occurrence
2 percent of row totals
3 percent of column totals
4 percent of table total

In Figure 3.24., lines 40 and 41, a weighted absolute occurrence table has been requested for each table variable.

From 1 to 4 occurrence statistics may be requested per run. All occurrence statistics numbers must appear as a contiguous block of cards immediately following the OCCURRENCE card. Tables will be written in the order in which the occurrence numbers appeared. One table will be printed for every occurrence statistic and table variable requested. The total number of tables produced is therefore the product of the number of occurrence statistics and the number of table variables.

3.6.2.8. SEARCH Card

The SEARCH command signals the end of a run stream and instructs PROGRAM SEARCH to search the Inventory Data File, retrieve the requested variable values, and perform the specified statistics. The command has no arguments, parameters, or specification lists. The SEARCH command begins in column 1.

3.6.2.9. END Card

The END command signals the end of all PROGRAM SEARCH runs, and no further information is read from the input file. The END command must begin in column 1.

3.6.2.10. Sample Input Form

A PROGRAM SEARCH sample input form is provided in Figure 3.25. for quick user reference to input formats and syntax.

FIGURE 3.25. PROGRAM SEARCH INPUT REFERENCE FORM

RUN NAME	64 character run title
COMMENT	input stream clarification comments
CONSTRAIN VAR	vcode 1
EXCLUDE IF	arg operand value
ROW GROUP	vcode 2
	class code 1 | strat class 1 | stratification class 1 description
ROW LIST	vcode 3
	class code 1 | stratification class 1 description
COL RANGE	vcode 4
	lower boundary | upper boundary | stratification class 1 description
TABLE VAR	vcode 5
TABLE VAR	vcode 6
STATISTICS	weight?
	stat 1
	stat n
OCCURRENCE	weight?
	occur 1
	occur n
SEARCH	
END	

55-6630-2

* U.S. GOVERNMENT PRINTING OFFICE 973-748/1535 33 REGION 10

3.6.3. Program Output

The input stream to PROGRAM SEARCH may include several run streams. The amount and type of output generated by the program depends upon the number of runs requested (each run separated by a SEARCH card) and number of table variables, occurrence statistics, and descriptive statistics specified in each run.

PROGRAM SEARCH prints a single title page for each input stream consisting of the program identification and date and time SEARCH was submitted for execution (Figure 3.26.). At the end of program execution, a closing page is written indicating the number of runs.

A complete set of output is written for every run stream within the program input stream. Some of the output generated by the sample run stream of Figure 3.24. is illustrated in Figures 3.27. through 3.31. Each set includes:

- A verbatim echo of the run stream (Figure 3.27.).

- A set of tables containing the descriptive or occurrence statistics of the table variables. If row and/or column stratifications were defined, the tables will be labeled with stratification variable headings and class headings (Figures 3.28. through 3.30.).

  -- A 1 page table will result from any stratification scheme using less than 10 column classes.

  -- A 2 page table will result from any stratification scheme using 10 or more column classes.

  -- The number of tables written per run stream is

  $$NTABLES = NTABVAR * (NSTATS + NOCCUR)$$

  where NTABLES is the number of tables written, NTABVAR is the number of TABLE VAR cards in the run stream, NSTATS is the number of STATISTICS card specifications, and NOCCUR is the number of OCCURRENCE card specifications.

- A description of the peripheral data file created by the run (Figure 3.31.).

FIGURE 3.26. PROGRAM SEARCH SAMPLE TITLE PAGE OUTPUT

```
                    NATIONAL FUELS INVENTORY LIBRARY

                    **** PROGRAM SEARCH ****
                    VERSION 01
                    LAST REVISED ON JUNE 22, 1979


                    DEVELOPED BY
                    SYSTEMS FOR ENVIRONMENTAL MGMT
                    P.O. BOX 3776
                    MISSOULA, MONTANA 59806


                    FOR THE
                    NATIONAL FUELS INVENTORY AND APPRAISAL PROJECT
                    ROCKY MOUNTAIN FOREST AND RANGE EXPERIMENT STATION
                    FORT COLLINS, COLORADO


                    BEGIN ON
                    DATE :   30 SEP 79
                    TIME :   21.25.54
```

FIGURE 3.27. PROGRAM SEARCH SAMPLE INPUT ECHO PAGE OUTPUT

```
RUN NAME        COVER TYPE VS STAND AGE
COMMENT         USE ALL FOREST SERVICE DATA
COMMENT         CONSTRAIN ELEVATION TO 3200 - 7400 FEET
CONSTRAIN VAR    301
    EXCLUDE IF    LT    3200
    EXCLUDE IF    GT    7400
COMMENT         COVER TYPE WILL BE THE ROW STRATIFICATION
ROW LIST         201
                    108            LODGEPOLE PINE
                    202            DOUGLAS-FIR
                    242            WESTERN RED CEDAR
COMMENT         STAND AGE WILL BE THE COLUMN STRATIFICATION
COL RANGE        211
                      1      50      1- 50
                     51     100     51-100
                    101     200    101-200
                    201     250    201-250
                    301     500    301-500
COMMENT         RETRIEVE 0.25 - 1.0 INCH FUEL LOAD
TABLE VAR        411
COMMENT         RETRIEVE 1.0 - 3.0 INCH FUEL LOAD
TABLE VAR        421
COMMENT         RETRIEVE GREATER THAN 3 INCH FUEL LOAD
TABLE VAR        431
COMMENT         RETRIEVE TOTAL DEAD AND DOWN FUEL LOAD
TABLE VAR        441
COMMENT         RETRIEVE DEAD AND DOWN FUEL INTERCEPT HEIGHT
TABLE VAR        501
COMMENT         RETRIEVE DEAD AND DOWN FUEL BULK DENSITY
TABLE VAR        501
COMMENT         DETERMINE SAMPLE SIZE, MEAN, AND ST DEV OF TABLE VARIABLES
COMMENT         NO WEIGHTING BY ENTRY PLOT SIZE
STATISTICS
                   1
                   4
                   6
COMMENT         DETERMINE SAMPLE SIZES IF WEIGHTED BY ENTRY PLOT SIZE
OCCURRENCE      WEIGHTED
                   1
COMMENT         END OF RUN 1
SEARCH
```

NATIONAL FUELS INVENTORY LIBRARY PROGRAM SEARCH (LAST REVISED ON 08 JUL 79) 30 SEP 79

RUN NO. 1 COVER TYPE VS STAND AGE

VARIABLE NO. 3 > 3.0 INCH DEAD AND DOWN FUEL LOAD (VARIABLE CODE 431)

TABLE NO. 2 SAMPLE SIZE

VARIABLE 211 STAND AGE

FOREST COVER TYPE (BY PLURALITY OF SPECI VARIABLE 201	1- 50	51-100	101-200	201-250	301-500	ROW TOTAL
LODGEPOLE PINE	10.	10.	10.	0.	0.	30.
DOUGLAS-FIR	10.	10.	10.	0.	0.	30.
WESTERN RED CEDAR	0.	0.	0.	0.	0.	0.
COLUMN TOTAL	20.	20.	20.	0.	0.	60.

RUN NO. 1 COVER TYPE VS STAND AGE
VAR NO. 3 > 3.0 INCH DEAD AND DOWN FUEL LOAD
TAB NO. 2 SAMPLE SIZE

1. 3. 2

PAGE NO. 10

FIGURE 3.28. PROGRAM SEARCH SAMPLE TABLE OUTPUT

FIGURE 3.29. PROGRAM SEARCH SAMPLE TABLE OUTPUT

NATIONAL FUELS INVENTORY LIBRARY PROGRAM SEARCH (LAST REVISED ON 08 JUL 79) 30 SEP 79

RUN NO. 1 COVER TYPE VS STAND AGE

VARIABLE NO. 3 > 3.0 INCH DEAD AND DOWN FUEL LOAD (VARIABLE CODE 431)

TABLE NO. 3 MEANS

VARIABLE 211 STAND AGE

FOREST COVER TYPE (BY PLURALITY OF SPECI VARIABLE 201	1- 50	51-100	101-200	201-250	301-500	ROW TOTAL
LODGEPOLE PINE	33.2208	20.9039	66.9492	-1.0000	-1.0000	57.0246
DOUGLAS-FIR	108.0154	35.1123	94.0377	-1.0000	-1.0000	79.0551
WESTERN RED CEDAR	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000
COLUMN TOTAL	95.6181	29.0081	80.4934	-1.0000	-1.0000	68.0399

NATIONAL FUELS INVENTORY LIBRARY PROGRAM SEARCH (LAST REVISED ON 08 JUL 79) 30 SEP 79

RUN NO. 1 COVER TYPE VS STAND AGE

VARIABLE NO. 3 > 3.0 INCH DEAD AND DOWN FUEL LOAD (VARIABLE CODE 431)

TABLE NO. 4 STANDARD DEVIATIONS

VARIABLE 211 STAND AGE

FOREST COVER TYPE (B Y PLURALITY OF SPECI VARIABLE 201	1- 50	51-100	101-200	201-250	301-500	ROW TOTAL
LODGEPOLE PINE	16.7800	7.2859	22.3186	-1.0000	-1.0000	31.2881
DOUGLAS-FIR	32.5973	15.0618	31.6882	-1.0000	-1.0000	41.7644
WESTERN RED CEDAR	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000
COLUMN TOTAL	28.2575	13.6285	30.0782	-1.0000	-1.0000	38.2351

RUN NO. 1 COVER TYPE VS STAND AGE
VAR NO. 3 > 3.0 INCH DEAD AND DOWN FUEL LOAD
TAB NO. 4 STANDARD DEVIATIONS

1. 3. 4 PAGE NO. 12

FIGURE 3.30. PROGRAM SEARCH SAMPLE TABLE OUTPUT

FIGURE 3.31. PROGRAM SEARCH SAMPLE PERIPHERAL DATA FILE STRUCTURE
OUTPUT

```
NFIL PROGRAM SEARCH PERIPHERAL DATA FILE STRUCTURE

RUN  1 RETRIEVED DATA ARE IN FILE  1 ON LOGICAL UNIT  4.

A TOTAL OF      427 RECORDS WERE WRITTEN TO THE FILE AS FOLLOWS:

RECORD TYPE   I:        1 RECORD INDICATING NUMBER OF REQUESTED TABLE VARIABLES TO BE RETRIEVED (  6)
RECORD TYPE  II:        6 RECORDS DOCUMENTING EACH REQUESTED TABLE VARIABLE
RECORD TYPE III:       50 RECORDS INDICATING EACH RETRIEVED INVENTORY ENTRY'S WEIGHT AND STRATIFICATION CLASS.
RECORD TYPE  IV:      350 RECORDS CONTAINING RETRIEVED TABLE VARIABLE VALUES FOR EACH ENTRY
                          (  6 REC TYPE IV'S FOLLOW EACH REC TYPE III)
```

3.6.4. Program Limitations

3.6.4.1. Lawrence Berkeley Laboratory CDC 6000/7000

The following limitations are in effect on the LBL CDC version of PROGRAM SEARCH:

- Maximum allowable variable code is 1500.
- Maximum number of variable codes on the Variable Index File is 500.
- Maximum number of CONSTRAIN VAR cards per run is 5.
- Maximum number of EXCLUDE IF specifications per CONSTRAIN VAR is 10.
- Maximum number of ROW or COL GROUP assignments is 20.
- Maximum number of ROW or COL stratification classes is 19.
- Maximum number of TABLE VAR cards (e.g., maximum number of table variables) is 10.

3.6.4.2. Fort Collins Computer Center UNIVAC 1100/82

The FCCC UNIVAC 1100 version of PROGRAM SEARCH has the following limitations:

- Maximum allowable variable code is 1500.
- Maximum number of variable codes on the Variable Index File is 500.
- Maximum number of CONSTRAIN VAR cards per run is 20.
- Maximum number of EXCLUDE IF specifications per is 100.
- Maximum number of ROW or COL GROUP assignments is 100.
- Maximum number of ROW or COL stratification classes is 19.
- Maximum number of TABLE VAR cards (e.g., maximum number of table variables) is 100.

3.6.5. Program Error Messages

PROGRAM SEARCH prints 14 error diagnostics to aid Library users in debugging program input streams. Most messages are triggered whenever any of the PROGRAM SEARCH limitations have been exceeded, or if any input command is unrecognizable. The message includes an error number, an error diagnosis, whether or not it was fatal, and the resulting fixups or assumptions made by the program.

All error messages are printed via a call to SUBROUTINE MESAGE from the routine detecting the error.

3.6.6. Program Structure and Logic

PROGRAM SEARCH consists of a main routine, 23 subroutines, a BLOCK DATA routine, and 3 labeled common blocks.

3.6.6.1. MAIN

- Step 1: Call SUBROUTINE HEADER to write title output page.
- Step 2: Call SUBROUTINE NEWRUN to initialize data constants and arrays.
- Step 3: Call SUBROUTINE INDEX to read and store variable codes, variable class codes, and base 10 exponent conversion factors from the Variable Index File on logical device N1.
- Step 4: Call SUBROUTINE READIN to read and interpret card input.
- Step 5: If any fatal errors were detected in Step 5, go to Step 2 and read next input run.
- Step 6: Call SUBROUTINE SEARCH to search for requested inventory information and write it to logical device N4.
- Step 7: Call SUBROUTINE OCCUR to compute and print retrieved variable occurrence statistics, if requested.
- Step 8: Call SUBROUTINE STATS to compute and print retrieved variable occurrence statistics, if requested.
- Step 9: Call SUBROUTINE PERIPH to print Peripheral Data File content summary for the run.
- Step 10: Return to Step 2 and read next input run. If last run, go to Step 11.
- Step 11: Call SUBROUTINE CLOSER to write closing output page.
- Step 12: STOP.

3.6.7. Logical Device Assignment

PROGRAM SEARCH uses 5 logical devices assigned to variables N1, N3,N4,N5, and N6 in the BLOCK DATA routine.

DEVICE	USE
N1	Variable Index File
N3	Inventory Data File
N4	Peripheral Data File
N5	Input
N6	Output

3.6.7.1. Lawrence Berkeley Laboratory CDC 6000/7000

The LBL CDC version of PROGRAM SEARCH assigns logical devices in the BLOCK DATA routine as follows:

```
DATA N1,N3,N4,N5,N6/1,3,4,5,6/
```

3.6.7.2. Fort Collins Computer Center UNIVAC 1100/82

The FCCC UNIVAC 1100 version of PROGRAM SEARCH assigns logical devices in the BLOCK DATA routine as follows:

```
DATA N1,N3,N4,N5,N6/2,3,4,5,6/
```

4. APPENDICES

APPENDICES

A. APPENDIX

A

A.1. N F I L STATUS SUMMARY : 01 OCTOBER 1979

The NFIL Fuels Inventory Database (FID) as of 1 October 1979 consists of:

13,429 inventory entries, assigned to

5,223 stands.

The initial FID inventory information was taken from three major data sets. The data sets were originally compiled by the following agencies or institutions:

RDA : Fire in Multiple Use Management RD&A Program, RWU 2108, USDA Forest Service Intermountain Forest & Range Experiment Station, Ogden, Utah. Northern Forest Fire Laboratory, Drawer G, Missoula, Montana 59806. Refer to Hartless, P. and R. Oberhue, unpublished internal office report.

CSU : Colorado State University Department of Forest & Wood Science, Fort Collins, Colorado. Refer to Pierce, Thomas L., 1977, unpublished masters thesis.

RØ1 : USDA Forest Service Region One forest inventory. No available reference or documentation.

The number of inventory entries and stands for each data set are summarized in Figure A.1. The RDA data consists of plot by plot inventory entries grouped into stands. The RØ1 data set consists entirely of stand summary inventory entries. The CSU data set consists of both plot by plot and stand summary entries.

Figures A.2. through A.5. stratifies the number of inventory entries by forest cover type and stand age class for the entire FID, and the individual RDA, CSU, and RØ1 data sets, respectively. Of the 13,429 entries, 9360 had both forest cover type and stand age information recorded.

FIGURE A.1. FUELS INVENTORY DATABASE STATUS SUMMARY

DATA SET	NUMBER OF INVENTORY ENTRIES	INVENTORY ENTRY KEYS FROM	TO
RDA	8052	1	8052
CSU	1020	8053	9072
R01	4357	9073	13429

DATA SET	NUMBER OF INVENTORY STANDS	STAND NUMBERS FROM	TO
RDA	831	1	831
CSU	35	832	866
R01	4357	867	5223

POOR QUALITY

The following pages are scanned at the best possible quality setting. The original pages are of poor quality and this is the best possible reproduction.

FIGURE A.2. FUELS INVENTORY DATABASE STATUS -- FOREST COVER TYPE

BY STAND AGE CLASS SAMPLE SIZE

NATIONAL FUELS INVENTORY LIBRARY PROGRAM SEARCH (LAST REVISED ON 08 JUL 79) 04 OCT 79

ROW NO. 1 INVENTORY OF A FUELS SUBSET - COVER TYPE BY STAND AGE OCCURRENCE

VARIABLE NO. 1 PRIMARY KEY FOR A SPECIFIC PLOT -- ASSIGNED BY SYSTEM (VARIABLE CODE 1)

TABLE NO. 1 SAMPLE SIZE

VARIABLE NO. STAND AGE ESTIMATE

GF, FIR=F, OTHER TYP E = SPECIES / VARIABLE NO.	1- 20	21- 50	51-100	101-150	151-200	201-250	251-300	301-350	351-999	*ROW TOTAL
MIXED CONIFER	0.	1.	2.	0.	0.	0.	0.	0.	0.	2.
FIR	0.	0.	0.	2.	0.	0.	0.	0.	0.	2.
DOUGLAS-FIR	4.	87.	555.	440.	355.	270.	145.	55.	10.	2437.
GRAND FIR	14.	44.	181.	45.	35.	10.	20.	0.	0.	351.
RED FIR	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
SUBALPINE FIR	42.	94.	251.	243.	183.	65.	24.	9.	13.	956.
WHITE FIR	0.	0.	1.	0.	21.	0.	0.	0.	0.	22.
OTHER	0.	0.	0.	1.	0.	0.	0.	0.	0.	1.
PONDEROSA PINE	44.	444.	1144.	729.	350.	97.	29.	10.	0.	3123.
PONDEROSA PINE	44.	84.	181.	342.	211.	80.	25.	5.	0.	888.
WHITE-BARK PINE	0.	6.	0.	22.	47.	52.	20.	3.	2.	172.
SPRUCE	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
SPRUCE - FIR	0.	69.	99.	74.	125.	89.	52.	15.	0.	463.
ENGELMANN SPRUCE	14.	6.	24.	41.	20.	9.	4.	0.	2.	135.
WESTERN LARCH	0.	4.	0.	0.	0.	0.	0.	0.	0.	15.
WESTERN RED CEDAR	14.	14.	33.	34.	33.	10.	24.	27.	48.	277.
LARCH - DOUGLAS FIR	23.	47.	74.	34.	24.	27.	10.	1.	0.	274.
GF - L - IF	14.	31.	79.	23.	34.	14.	5.	2.	0.	196.
L - SS - IF	0.	4.	12.	7.	5.	4.	0.	0.	0.	36.
COLUMN TOTAL	1 [illegible]	[illegible]	[illegible]	2174.	1444.	733.	356.	133.	75.	9360.

ROW NO. 1 INVENTORY DATA FILES SUMMARY - COVER TYPE BY STAND AGE OCCURRENCE
VAR NO. 1 PRIMARY KEY TO FIA INVENTORY PLOT -- ASSIGNED BY SYSTEM
TAB NO. 1 SAMPLE SIZE

POOR QUALITY

The following pages are scanned at the best possible quality setting. The original pages are of poor quality and this is the best possible reproduction.

FIGURE A.3. RDA FUELS SET SUMMARY -- FOREST COVER TYPE

BY STAND AGE CLASS SAMPLE SIZE

NATIONAL FUEL- INVENTORY LIBRARY PROGRAM SEARCH (LAST REVISED ON 08 JUL 79) 02 OCT 79

RUN NO. 1 RDA FUELS SUMMARY - COVER TYPE BY STAND AGE SAMPLE SIZE

VARIABLE NO. 1 PRIMARY KEY ID FOR INVENTORY PLOT -- ASSIGNED BY SYSTEM (VARIABLE CODE 1)

TABLE NO. 1 SAMPLE SIZE

VARIABLE 304 STAND AGE ESTIMATE

GEN FOREST COVER TYP E - SPECIES VARIABLE 303	1- 20	21- 50	51-100	101-150	151-200	201-250	251-300	301-350	351-999	RDA TOTAL
MIXED CONIFER	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
FIR	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
DOUGLAS-FIR	0.	135.	261.	194.	357.	161.	107.	35.	0.	1251.
GRAND FIR	10.	19.	159.	95.	35.	10.	20.	0.	0.	348.
RED FIR	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
SUBALPINE FIR	16.	12.	177.	138.	92.	12.	3.	3.	9.	458.
WHITE FIR	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
PINE	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
LODGEPOLE PINE	23.	274.	737.	517.	461.	68.	18.	13.	0.	2117.
PONDEROSA PINE	10.	3.	44.	340.	133.	74.	5.	0.	0.	616.
WHITEBARK PINE	0.	14.	0.	3.	35.	47.	18.	3.	2.	131.
SPRUCE	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
SPRUCE - FIR	0.	63.	84.	74.	125.	89.	52.	15.	0.	468.
ENGELMANN SPRUCE	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
WESTERN LARCH	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
WESTERN RED CEDAR	10.	1.	20.	10.	40.	0.	20.	25.	45.	171.
LARCH - DOUGLAS FIR	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
GF - L - DF	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
L - SP - DF	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
COLUMN TOTAL	67.	721.	1481.	1379.	1278.	461.	243.	94.	56.	5560.

RUN NO. 1 RDA FUELS SUMMARY - COVER TYPE BY STAND AGE SAMPLE SIZE
VAR NO. 1 PRIMARY KEY ID FOR INVENTORY PLOT -- ASSIGNED BY SYSTEM
TAB NO. 1 SAMPLE SIZE

1. 1. 1 PAGE NO. 1

POOR QUALITY

The following pages are scanned at the best possible quality setting. The original pages are of poor quality and this is the best possible reproduction.

FIGURE A.4. CSU FUELS SET SUMMARY -- FOREST COVER TYPE BY STAND AGE CLASS SAMPLE SIZE

NATIONAL FUEL INVENTORY LIBRARY PROGRAM SEARCH (LAST REVISED ON 08 JUL 79) 02 OCT 79

RUN NO. 1 CSU FUELS DATA SUMMARY - COVER TYPE BY STAND AGE SAMPLE SIZE

VARIABLE NO. 1 PRIMARY KEY ID FOR INVENTORY PLOT -- ASSIGNED BY SYSTEM (VARIABLE CODE 1)

TABLE NO. 1 SAMPLE SIZE

VARIABLE 304 STAND AGE ESTIMATE

GEN FOREST COVER TYP E - SPECIES VARIABLE 303	1- 20	21- 50	51-100	101-150	151-200	201-250	251-300	301-350	351-999	ROW TOTAL
MIXED CONIFER	0.	0.	2.	0.	0.	0.	0.	0.	0.	2.
FIR	0.	0.	0.	2.	0.	0.	0.	0.	0.	2.
DOUGLAS-FIR	1.	4.	13.	103.	4.	5.	0.	0.	0.	130.
GRAND FIR	0.	0.	2.	1.	0.	0.	0.	0.	0.	3.
RED FIR	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
SUBALPINE FIR	0.	0.	9.	35.	0.	0.	0.	0.	0.	45.
WHITE FIR	0.	0.	1.	0.	21.	0.	0.	0.	0.	22.
PINE	0.	0.	0.	1.	0.	0.	0.	0.	0.	1.
LODGEPOLE PINE	1.	2.	21.	32.	0.	3.	1.	0.	0.	60.
PONDEROSA PINE	0.	1.	7.	8.	38.	0.	0.	0.	0.	54.
WHITEBARK PINE	0.	0.	0.	1.	0.	0.	0.	0.	0.	1.
SPRUCE	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
SPRUCE - FIR	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
ENGELMANN SPRUCE	0.	0.	0.	17.	0.	0.	0.	0.	0.	17.
WESTERN LARCH	0.	15.	0.	0.	0.	0.	0.	0.	0.	15.
WESTERN RED CEDAR	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
LARCH - DOUGLAS FIR	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
GF - L - DF	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
L - SP - DF	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
COLUMN TOTAL	2.	22.	55.	201.	64.	8.	1.	0.	0.	352.

RUN NO. 1 CSU FUELS DATA SUMMARY - COVER TYPE BY STAND AGE SAMPLE SIZE
VAR NO. 1 PRIMARY KEY ID FOR INVENTORY PLOT -- ASSIGNED BY SYSTEM
TAB NO. 1 SAMPLE SIZE

1. 1. 1 PAGE NO. 1

POOR QUALITY

The following pages are scanned at the best possible quality setting. The original pages are of poor quality and this is the best possible reproduction.

FIGURE A.5. R01 FUELS SET SUMMARY -- FOREST COVER TYPE

BY STAND AGE CLASS SAMPLE SIZE

NATIONAL FUEL INVENTORY LIBRARY PROGRAM SEARCH (LAST REVISED ON 08 JUL 79) 02 OCT 79

ROW NO. 1 REGION 1 FUELS SUMMARY - COVER TYPE BY STAND AGE SAMPLE SIZE

VARIABLE NO. 1 PRIMARY KEY (PLOT INVENTORY PLOT -- ASSIGNED BY SYSTEM (VARIABLE CODE 1)

TABLE NO. 1 SAMPLE SIZE

VARIABLE NO. 2 STAND AGE ESTIMATE

GEN FOREST COVER TYP E - SPECIES VARIABLE NO 3	1- 20	21- 50	51-100	101-150	151-200	201-250	251-300	301-350	351-999	ROW TOTAL
MIXED CONIFER	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
FIR	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
DOUGLAS-FIR	47.	125.	254.	143.	204.	104.	38.	20.	10.	1056.
GRAND FIR	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
RED FIR	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
SUBALPINE FIR	33.	49.	49.	94.	91.	53.	21.	6.	4.	453.
WHITE FIR	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
PINE	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
LODGEPOLE PINE	78.	154.	341.	175.	99.	25.	10.	3.	0.	945.
PONDEROSA PINE	18.	62.	55.	44.	40.	12.	18.	5.	0.	218.
WHITEBARK PINE	0.	2.	5.	13.	12.	5.	2.	0.	0.	40.
SPRUCE	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
SPRUCE - FIR	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
ENGELMANN SPRUCE	13.	40.	24.	24.	20.	9.	4.	0.	2.	121.
WESTERN LARCH	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
WESTERN RED CEDAR	5.	15.	15.	24.	25.	10.	4.	2.	3.	106.
LARCH - DOUGLAS FIR	11.	47.	74.	59.	24.	27.	10.	1.	0.	274.
DF - L - DF	27.	61.	73.	23.	33.	14.	5.	2.	0.	198.
L - DF - DF	0.	5.	12.	7.	5.	4.	0.	0.	0.	35.
COLUMN TOTAL	337.	741.	1044.	602.	557.	264.	112.	39.	19.	3445.

ROW NO. 1 REGION 1 FUELS SUMMARY - COVER TYPE BY STAND AGE SAMPLE SIZE
VAR NO. 1 PRIMARY KEY (PLOT INVENTORY PLOT -- ASSIGNED BY SYSTEM
TAB NO. 1 SAMPLE SIZE

1. 1. 1

B. APPENDIX

B

C. APPENDIX

C

D. APPENDIX

D

E. APPENDIX

E

E. RECOMMENDED VARIABLE INDEX FILE CODES

The Variable Index File documents all inventory variable codes and units of measure used by the Fuels Inventory Database (discussed in Section 2.3.) Variable code numbers represent four general types of fuel inventory information.

- Location Information — Code Variables 007 - 099
- Site Information — Code Variables 100 - 299
- Stand Information — Code Variables 300 - 699
- Fuel Information — Code Variables 700 - 1499

As new data is submitted to the Fuels Inventory Database, new variable codes and variable class codes may have to be added to the Variable Index File (discussed in Section 3.1.) The Management Software Package allows the user the flexibility to assign any number to a variable code as long as that code number does not already appear on the current Variable Index File. To aid in assigning numbers to new variable codes, Figure E.1 gives a recommended structure for coding new location, site, or stand information entries into the Variable Index File. For a complete listing of all codes currently on the system see Appendix G, Section G.1, Variable Index File.

Variable codes which describe fuel information (code variables 0700 - 1499) are a four digit concatonated key. The first two digits identify the particular fuel characteristic measured (eg. 07-- Load, 09-- Highest intercept depth). The last two digits digits identify the particle type being measured and indicate whether the variable code value that follows is an actual fuel measurement or a collection method code (eg. 0701 Load litter collection method, 0751 Load litter ton/acre). All collection method codes are ordinal and all fuel measurement codes are continuous. It is recommended that all new fuel information variable entries are coded according to the numbering system presented in Figure E.2. For a complete listing of all codes on the system see Appendix G, Section G.1, Variable Index File.

FIGURE E.1 : RECOMMENDED VARIABLE INDEX FILE CODES

LOCATION, SITE, AND STAND VARIABLES

CODE TYPE	CODE NUMBER	CLASS NUMBER	VARIABLE DESCRIPTION
cont	70	9999	UTM 1
ordl	71		UTM 1 Reference
		1	North
		2	East
		3	South
		4	West
cont	72	9999	UTM 2
ordl	73		UTM 2 Reference
		1	North
		2	East
		3	South
		4	West
cont	90	9999	Longitude
ordl	91		Longitude Reference
		1	North
		2	East
		3	South
		4	West
cont	92	9999	Latitude
ordl	93		Latitude Reference
		1	North
		2	East
		3	South
		4	West
ordl	101		Land/Vegetation Condition Class
		1	Bog
		2	Marsh
		3	Meadow
		4	Prairie
		5	Grassland
		6	Sage - scrub
		7	Shrub
		8	Alpine zone - Krummholtz
		9	Upright forest

CODE TYPE	CODE NUMBER	CLASS NUMBER	VARIABLE DESCRIPTION
ordl	106		Topography
		1	Level
		2	Draw - Ravine
		3	Slope
		4	Ridgetop
		5	Peak
ordl	107		Natural Site Damage/Disturbance Type
		1	Disturbed -- type unknown
		2	Avalanche
		3	Land/mud slide
		4	Flooding
		5	Erosion - water
		6	Erosion - wind
cont	150	9999	Site Productivity (cu/ft/ac)
cont	151	9999	Site Productivity (cu/ft)
cont	152	9999	Site Productivity (CCF)
	.		
	.		
	.		
cont	169	9999	Site Productivity (mean annual increment)
cont	200	9999	Site Index - Spp Unknown (ht at 50 years)
cont	201	9999	Site Index - Spp Unknown (ht at 100 years)
cont	202	9999	Site Index - Douglas fir (ht at 50 years)
cont	203	9999	Site Index - Douglas fir (ht at 100 years)
cont	204	9999	Site Index - W. Larch (ht at 50 years)
cont	205	9999	Site Index - W. Larch (ht at 100 years)
	.		
	.		
	.		
cont	250	9999	Site Index - Black Oak (ht at 50 years)
cont	251	9999	Habitat Type - Pfister, Habitat Types of Montana
cont	252	9999	Habitat Type - Daubenmire
	.		
	.		
	.		
cont	260	9999	Habitat Type - Author, Location

CODE TYPE	CODE NUMBER	CLASS NUMBER	VARIABLE DESCRIPTION
ord1	301		Stand Structure
		1--	No overstory component
		2--	Even aged overstory
		3--	Two storied stand
		4--	Three storied stand
		-00	Unknown forest floor
		-01	Grass forest floor
		-02	Herb forest floor
		-03	Low shrub forest floor
		-04	High shrub forest floor
		-05	Shrub forest floor
		-06	Regeneration forest floor (tree spp only)
cont	305	9999	Years Since Major Burn Estimate
cont	311	9999	Stand Avg DBH
cont	312	9999	Stand Avg DBH Live Stems
cont	313	9999	Stand Avg DBH Dead Stems
cont	321	9999	Stand Avg Density (stem/ac)
cont	322	9999	Stand Avg Density Live Stems
cont	323	9999	Stand Avg Density Dead Stems
cont	331	9999	Stand Avg Basal Area
cont	332	9999	Stand Avg Basal Area Live Stems
cont	333	9999	Stand Avg Basal Area Dead Stems
cont	341	9999	Stand Avg Volume per Acre
cont	342	9999	Stand Avg Volume per Acre Live Stems
cont	343	9999	Stand Avg Volume per Acre Dead Stems
cont	352	9999	Crown Length
cont	353	9999	Crown Width
cont	356	9999	Crown Competition Factor

CODE TYPE	CODE NUMBER	CLASS NUMBER	VARIABLE DESCRIPTION
ordl	381		Natural Stand Disturbance/Damage Type
		1	Fire - not prescribed
		2	Insect
		3	Hydric - Flooding - water damage
		4	Disease
		5	Windthrow
		6	Snow/ice
		7	Avalanche
		8	Combined alpine effects
		9	Wildlife
		10	Domestic animals
		11	Animals - type unknown
		12	Weather effects
cont	382	9999	Natural Stand Damage/Disturbance Intensity in Percent
cont	383	9999	Nat Stand Dam/Dist Period (elapsed time since)
ordl	384		Management Activities in Stands
		001	Activity indicated - type unknown
		1--	Regeneration Cuts
		-00	unknown
		-01	patch
		-02	strip
		-03	stand
		2--	Seedtree Cuts
		-00	unknown
		-01	preperation cut
		-02	seed cut/harvest
		-03	overstory removal cut
		3--	Shelterwood Cuts
		-00	unknown
		-01	preperation cut step 1
		-02	seed cut/harvest
		-03	overstory removal cut
		4--	Selection Cuts
		-00	unknown
		-01	individual tree selection
		-02	group selection
		5--	Intermediate Cuts
		-00	unknown
		-01	commercial thinning
		-02	salvage
		6--	Release Cuts
		-00	unknown
		-01	individual tree selection
		-02	group selection
		7--	Precommercial Thinnings
		-00	unknown
		-01	individual select trees
		-02	strip
		-03	area
		-04	pruning

CODE TYPE	CODE NUMBER	CLASS NUMBER	VARIABLE DESCRIPTION
cont	385	9999	Management Activities in Stands - Time Period
ordl	386		Mgt Activity on Residual Stands/Sites - Type
		10	Site preperation indicated - type unknown
		11	Prescribed burn
		12	Lopping/slash
		13	Jackpot burn/slash
		14	Machine pile/slash
		15	Machine terrace
		16	Machine scarify
		17	Machine furrow, disc
		18	Hand pile/slash
		19	Rolling,chopping,trampling/ slash
		20	Herbacide spraying
		50	Reforestation indicated - type unknown
		51	Natural reforestation
		52	Seeding reforestation
		53	Planting reforestation
cont	387	9999	Mgt Activity on Residual Stand/Site - Time Period
ordl	388		Bolewood Removal Practices
		1	Indicated - type unknown
		2	Machine skidding
		3	High lead (log elevated one end)
		4	Aerial skidding (helicopter etc)
	400		
	.		
	.		(Variable Codes - First Component Same as Stand Codes)
	.		
cont	406	9999	Percent of stand in First Component
	.		
	.		
	.		
	499		
	500		
	.		
	.		(Variable Codes - Second Component Same as Stand Codes)
	.		
	599		
	600		
	.		
	.		(Variable Codes - Third Component Same as Stand Codes)
	.		
	699		

FIGURE E.2 : RECOMMENDED VARIABLE INDEX FILE CODES

FUEL INFORMATION CODES

CODE NUMBER	MFACTOR 1.0E N	VARIABLE DESCRIPTION
		FUEL CHARACTERISTICS MEASURED
07--	-3	Load
08--	-1	Average Depth
09--	-1	Highest Intercept Depth
10--	-2	Percent Cover
11--	-2	Percent Live
12--	-4	Bulk Density
		PARTICLE TYPE - COLLECTION METHOD
--01		Litter
--02		Duff
--03		Tree seedlings/regeneration
--04		Cones
--05		Bark scales
--10		Down woody fuels total
--11		" " " total .LE. 3" dia
--12		" " " total .GT. 3" dia
--13		" " " 0 - ¼" dia
--14		" " " ¼ - 1" dia
--15		" " " 1 - 3" dia
--16		" " " + 3" dia sound only
--17		" " " + 3" dia rotten only
--30		Shrub fuels total
--31		" " total stemwood
--32		" " foliage
--33		" " small stemwood (0.0 - 2.0 cm dia)
--34		" " large stemwood (.GT. 2.0 cm dia)
--40		Grass and herbs total
--41		" " " live only
--42		" " " dead only

CODE NUMBER	MFACTOR 1.0E N	VARIABLE DESCRIPTION
		PARTICLE TYPE - MEASURED
--51		Litter
--52		Duff
--53		Tree seedlings/regeneration
--54		Cones
--55		Bark Scales
--60		Down woody fuels total
--61		" " " total .LE. 3" dia
--62		" " " total .GT. 3" dia
--63		" " " 0 - ¼" dia
--64		" " " ¼ - 1" dia
--65		" " " 1 - 3" dia
--66		" " " + 3" dia sound only
--67		" " " + 3" dia rotten only
--80		Shrub fuels total
--81		" " total stemwood
--82		" " foliage
--83		" " small stemwood (0.0 - 2.0 cm dia)
--84		" " large stemwood (.GT. 2.0 cm dia)
--90		Grass and herbs total
--91		" " " live only
--92		" " " dead only

EXAMPLES:

0701	Load litter collection method
0751	Load litter
1141	Percent cover live grass and herbs collection method
1191	Percent cover live grass and herbs
1260	Bulk density total down woody fuels

F. APPENDIX

F

F. M S P PROGRAM LISTINGS

Listings of the 6 Management Software Package programs are appended in accompanying computer printout binders. Each listing is complete with a sample output corresponding to the input and output illustrations in Sections 2 and 3.

F.1. PROGRAM INDEX

F.2. PROGRAM DATA

F.3. PROGRAM COUNT

F.4. PROGRAM LIST

F.5. PROGRAM REVIEW

F.6. PROGRAM SEARCH

G. APPENDIX

G

G. F I D FILE LISTINGS

Listings of the Fuels Inventory Database Files as per the FID status (Appendix A.) are appended in accompanying computer printout binders. The file listings are:

G.1. VARIABLE INDEX FILE

G.2. STAND DOCUMENTATION FILE

The Inventory Database File, consisting of some 16 million bytes, is not included in the file listings.